

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет електроніки

(повна назва інституту/факультету)

Кафедра звукотехніки та реєстрації інформації

(повна назва кафедри)

«На правах рукопису»

УДК 004.738.5

«До захисту допущено»

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ ” 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 171 Електроніка (Електронні та інформаційні технології кінематографії та аудіовізуальних систем)

(код і назва спеціальності)

на тему: «Особливості використання “хмарної” платформи OpenStack для мережі цифрових кінотеатрів».

Виконав студент VI курсу, групи ДВ-71мп

(шифр групи)

Шпула Олександр Андрійович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник доцент, к.т.н. Трапезон К.О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

(назва розділу)

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Інститут (факультет) _____ Факультет електроніки _____
(повна назва)

Кафедра _____ Кафедра звукотехніки та реєстрації інформації _____
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (спеціалізація) 171 Електроніка (Електронні та інформаційні технології кінематографії та аудіовізуальних систем)
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Шпули Олександра Андрійовича**
(прізвище, ім'я, по батькові)

1. Тема дисертації Особливості використання “хмарної” платформи OpenStack для мережі цифрових кінотеатрів.

науковий керівник дисертації Трапезон Кирил Олександрович к.т.н., _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «07» листопада 2018р. №4114-с

2. Строк подання студентом дисертації 10.12.2018р. _____

3. Об'єкт дослідження: мережа глядацьких залів цифрового кінотеатру формату D-Cinema

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) 1) базова “хмарна” платформа – Open Stack; технологія цифрового кіно – D-Cinema; кількість глядацьких залів мережі кінотеатрів – не менше 5; наявність багаторівневої безпеки передавання даних – так; формат кінофільмів з роздільною здатністю – 2K або 4K; базова система кіно проєкції – Dolby Digital Cinema.

5. Перелік завдань, які потрібно розробити Визначення передумов до впровадження хмарної платформи OpenStack в інформаційних системах, які характеризуються зберіганням, передаванням та конвертацією великих

об'ємів інформації; аналіз типової структурної схеми створення багатозального кінотеатру, визначення шляхів розширення кінотеатру до мережі. Розгляд та виявлення слабких місць у функціонуванні сучасних кінотеатрів формату D-Cinema; розробка рекомендацій з перероблення існуючої схеми взаємодії основних ланок кінотеатру Dolby D-Cinema з метою впровадження у їх роботу хмарних сервісів.

6. Перелік графічного (ілюстративного) матеріалу 15 рис., 19 табл., 1 презентація, 10 слайдів.

7. Орієнтовний перелік публікацій «Хмарні технології для проектування сучасних кінотеатрів на базі Openstack», «Реалізація хмарних технологій у сучасному проектуванні цифрових кінотеатрів».

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 10.09.2017

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
	Написання першого розділу: «Платформа Openstack».	10.10.2017	
	Написання другого розділу: «Особливості створення мережі digital cinema».	15.12.2017	
	Написання третього розділу: «Особливості віртуалізації мережі цифрового кінотеатру».	01.05.2018	
	Написання четвертого розділу: «Розроблення стартап-проекту».	09.11.2018	
	Підготовка матеріалів до друку та оформлення пояснювальної записки	25.11.2018	
	Підготовка та оформлення презентації для доповіді	30.11.2018	

Студент

(підпис)

О.А. Шпула

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

К. О. Трапезон

(ініціали, прізвище)

УДК 004.738.5

РЕФЕРАТ

Магістерська дисертація: 78 с., 15 рис., 19 табл., 7 джерел.

КІНОТЕАТР, СЕРВЕР КІНОПОКАЗУ, ХМАРНІ ОБЧИСЛЕННЯ,
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЗАХИСТ, КЛЮЧ.

Актуальність роботи полягає у тому, що на сьогоднішній день мережні технології у тому числі в цифровому кіно набувають все більшої популярності, зокрема для відтворення та передачі візуального контенту, а концепція хмарних обчислень є однією з найбільш перспективних тенденцій розвитку в сфері інформаційних технологій.

Об'єктом дослідження є мережа глядацьких залів цифрового кінотеатру формату D-Cinema.

Метою роботи є розробка рекомендацій задля подальшої реалізації та впровадження технічного рішення з проектування мережі гібридних цифрових кінотеатрів, де б з урахуванням впровадження обраної “хмарної” платформи можна було б досягти підвищення гнучкості, надійності та захисту цифрових мастер-копій при функціонуванні кінотеатру формату D-Cinema.

Для досягнення поставленої мети необхідно виконати такі завдання:

- визначення передумов до впровадження хмарної платформи OpenStack в інформаційних системах, які характеризуються зберіганням, передаванням та конвертацією великих об'ємів інформації;
- аналіз типової структурної схеми створення багатозального кінотеатру, визначення шляхів розширення кінотеатру до мережі. Розгляд та виявлення слабких місць у функціонуванні сучасних кінотеатрів формату D-Cinema;
- розробка рекомендацій з перероблення існуючої схеми взаємодії основних ланок кінотеатру Dolby D-Cinema з метою впровадження у їх роботу хмарних сервісів.

SUMMARY

Master's dissertation: 78 p., 15 pic., 19 tabl., 7 sources.

CINEMATOGRAPHY, SERVER OF CINOPACK, CLEANING CALCULATIONS, SOFTWARE, PROTECTION, KEY.

The urgency of the work is that to date, network technologies, including digital cinema, are gaining in popularity, in particular for the reproduction and transmission of visual content, and the concept of cloud computing is one of the most promising trends in the field of information technology.

The object of the study is a network of auditoriums of the digital cinema format D-Cinema.

The aim of the work is to develop recommendations for the further implementation and implementation of technical solutions for the design of a network of hybrid digital cinemas, where, taking into account the implementation of the selected "cloud" platform, it would be possible to achieve increased flexibility, reliability and protection of digital master copies in the operation of the cinema format D-Cinema.

To achieve the goal you must accomplish the following tasks:

- Determining the preconditions for implementing the OpenStack cloud platform in information systems that are characterized by storing, transmitting and converting large volumes of information;
- analysis of the typical structural scheme of creating a multi-channel cinema, determining the ways to expand the cinema to the network. Consideration and identification of weaknesses in the functioning of modern cinemas in the D-Cinema format;
- development of recommendations for the redesign of the existing scheme of interaction of the main parts of the Dolby D-Cinema cinema with a view to implementing cloud services in their work.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП.....	8
1 ПЛАТФОРМА OPENSTACK.....	10
1.1 Поняття концепції Openstack. Типи “хмар”.....	10
1.2 Архітектура OpenStack.....	16
1.2.1 Сервіс Swift як образ кіно накопичувача.....	20
1.2.2 Організація вузлів зберігання сервісу Swift.....	22
1.3 Захист даних в “хмарних” системах.....	26
2 ОСОБЛИВОСТІ СТВОРЕННЯ МЕРЕЖІ DIGITAL CINEMA.....	31
2.1 Особливості створення мережі кінотеатрів D-cinema.....	31
2.2 Система управління кінотеатром.....	37
3 ОСОБЛИВОСТІ ВІРТУАЛІЗАЦІЇ МЕРЕЖІ ЦИФРОВОГО КІНОТЕАТРУ.....	41
3.1 Передумови переходу до нових технологій в кіно.....	41
3.2 Особливості захисту кінофільмів в кінотеатрах	46
3.3 Технічні рішення з впровадження платформи Open Stack в цифрових кінотеатрах.....	48
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	55
4.1 Опис ідеї проекту.....	55
4.2 Технологічний аудит ідеї проекту.....	57
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	58
4.4 Розроблення ринкової стратегії проекту.....	62
4.5 Розроблення маркетингової програми стартап-проекту.....	64
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	70
ДОДАТОК А.....	71

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

BM	–	Віртуальна машина;
OC	–	Операційна система;
СЗД	–	Система зберігання даних;
ЦОД	–	Центр обробки даних;
AES	–	Advanced Encryption Algorithm;
DCI	–	Digital Cinema Initiatives;
DLP	–	Digital Light Processing;
IaaS	–	Infrastructure as a Service;
IMB	–	Integrated Media Block;
KDM	–	Key Delivery Management;
KVM	–	Kernel-based Virtual Machine;
PaaS	–	Platform as a Service;
SaaS	–	Software as a Service;
SDS	–	Software-Defined Storage;
TMS	–	Theatre Management System.

ВСТУП

В умовах стрімкого розвитку розважальних послуг у світі, ледь не на перше місце виходить питання створення кінотеатрального комплексу, де окрім цифрового глядацького залу і традиційного кінопоказу, відвідувачам пропонуються і інші розважальні заходи – боулінг, картинг, ковзанка тощо. Натомість, питання функціонування багатозальної мережі містить ще і суто технічні труднощі, які пов'язані з забезпеченням ліцензій кіно продукту при прокаті кінофільмів. Так можна стверджувати, що кінодистриб'ютери вже не в повній мірі мають право довіряти власникам кінотеатрів з точки зору забезпечення останніми захисту кінофільмів, особливо коли відбуваються прем'єри даної картини по усьому світу. Тобто, питання захисту інтелектуальних прав вже не відіграє сьогодні другу роль при демонстрації кінофільмів. Шляхом до підвищення захисту контенту може стати впровадження концепції OpenStack, яка дозволяє відокремити користувача послуги (глядача, кіномеханік, інженер) і власника продукту. З означених міркувань, вважаю обрану тему **актуальною**.

Метою дослідження є розробка рекомендацій задля подальшої реалізації та впровадження технічного рішення з проектування мережі гібридних цифрових кінотеатрів, де б з урахуванням впровадження обраної “хмарної” платформи можна було б досягти підвищення гнучкості, надійності та захисту цифрових мастер-копій при функціонуванні кінотеатру формату D-Cinema.

Для досягнення мети були поставлені такі завдання:

- визначення передумов до впровадження хмарної платформи OpenStack в інформаційних системах, які характеризуються зберіганням, передаванням та конвертацією великих об'ємів інформації;
- аналіз типової структурної схеми створення багатозального кінотеатру, визначення шляхів розширення кінотеатру до мережі. Розгляд та

виявлення слабких місць у функціонуванні сучасних кінотеатрів формату D-Cinema;

– розробка рекомендацій з перероблення існуючої схеми взаємодії основних ланок кінотеатру Dolby D-Cinema з метою впровадження у їх роботу хмарних сервісів.

Методом дослідження є зв'язки та принципи взаємодії технічних ланок цифрового кінотеатру з точки зору передавання цифрових DCP-пакетів, їх алгоритмів захисту та відтворення у розрізі впровадження обраної незалежної “хмарної” платформи.

Об'єктом дослідження є мережа глядацьких залів цифрового кінотеатру формату D-Cinema.

Новизна дослідження полягає у розробленні ідеї щодо модернізації існуючих кінотеатрів формату D-Cinema задля забезпечення останніми можливості значного розширення об'єму кінонакопичувачів з функціями захисту через переведення в режим віртуалізації функціонування.

Практична цінність полягає у тому, що розроблені підходи та принципи дозволять у майбутньому розробити нову галузь у проектуванні гібридних цифрових кінотеатрів формату D-Cinema за умови дотримання існуючих міжнародних рекомендацій DCI.

1 ПЛАТФОРМА OPENSTACK

1.1 Поняття концепції Openstack. Типи “хмар”

Хмарні обчислення - це модель надання широко доступного, зручного доступу по мережі до загального пулу налаштованих обчислювальних ресурсів на вимогу. До таких ресурсів можна віднести: мережі, сервери, системи зберігання даних, додатки і сервіси. Оскільки сучасний цифровий кінотеатр за схемою підключень основних складових є фактично інформаційною мережею, то цілком зрозуміло, що така модель обчислень може бути запроваджена у роботу цієї мережі. Розрізняють три основні сервісні моделі хмари в межах концепції OpenStack: SaaS (Software as a Service - «програма як сервіс»), IaaS (Infrastructure as a Service - «інфраструктура як сервіс») і PaaS (Platform as a Service - «платформа як сервіс») [1].

Інфраструктура як сервіс (IaaS) - сервісна модель OpenStack, яку використовують в основному саме для розгортання хмар. В даному випадку користувач отримує контроль за усіма рівнями стека програмного забезпечення, що лежать вище хмарної платформи, а саме: віртуальними машинами, мережами, виділеним користувачеві обсягом простору на системі зберігання даних (СЗД). У цьому випадку користувач виступає адміністратором операційної системи і всього, що працює поверх, аж до додатків. Прикладами платформ, які забезпечують подібну модель, крім OpenStack, можна назвати Apache CloudStack, Eucalyptus і OpenNebula.

Платформа, як сервіс (PaaS) - хмара, яка побудована за такою моделлю, цілком може розташовуватись «всередині» хмари моделі IaaS. В цьому випадку межа контролю користувача лежить на рівні платформи побудови додатків, наприклад сервера додатків, бібліотек, середовища розробки або бази даних. Користувач не контролює і не адмініструє віртуальні машини і операційні системи, встановлені на них. Приклади хмарних платформ моделі PaaS: Apache Stratos, Cloud Foundry, Deis і OpenShift Origin.

Програмне забезпечення як сервіс (SaaS) - в цьому випадку межа контролю користувача - сам додаток. Користувач в даному випадку може навіть не знати, що таке віртуальна машина або операційна система, він просто працює з додатком. Приклади таких хмарних продуктів: Google Docs, Office 365.

Для кожної моделі може бути чотири типи реалізації “хмари”: публічні (public); приватні (private); громадські (community), призначені для конкретної спільноти споживачів; і гібридні (Hybrid cloud).

Приватна хмара - вся інфраструктура розгорнута в центрі обробки даних (ЦОД) і служить підрозділом однієї компанії або групи компаній.

Публічна хмара - замовником хмарних послуг може виступати будь-яка компанія або навіть приватна особа. Це модель впровадження, на якій заробляють провайдери хмарних послуг.

У публічній хмарі ІТ-сервіси та інфраструктурні ресурси надаються поверх глобальної мережі інтернет. Можливість використання інфраструктури стороннього провайдера створює безліч можливостей для ефективного завантаження ресурсів та їх перерозподілу. Великий мінус полягає в тому, що публічні хмари набагато більш уразливі в порівнянні з приватними.

Оскільки сервери додатків знаходяться в публічній хмарі, виключається ризик простою бізнес-процесів через серверні аварії. Віртуальні сервери провайдерів найчастіше налаштовані на потужній фізичній базі, розміщені у великих дата-центрах, де можливий час простоїв обчислюється хвилинами в рік. Використання публічних хмар і відсутність контакту користувачів зі складним комп'ютерним обладнанням дозволяє відмовитися від послуг ІТ-фахівців у конкретній організації.

Натомість, головним мінусом публічної хмари є відсутність можливостей контролю з боку організації: працездатність послуг повністю підпорядкована сторонньому провайдеру. Крім цього, продуктивність публічних хмарних сервісів безпосередньо залежить від стабільності і смуги

пропускання інтернет-з'єднання, в деяких випадках передачі даних може бути досить повільним. При оперуванні великими обсягами даних публічні хмари не витримують конкуренції з приватними по продуктивності. Слабка захищеність даних - ще одна характерна риса публічних хмарних середовищ. Якби зусилля не робив провайдер в області забезпечення безпеки, захист приватної хмари завжди буде на порядок більш надійною. І ця обставина обмежує їх використання при розгортанні сучасних цифрових кінотеатрів формату D-Cinema. На рисунку 1.1 показано відмінності між двома варіантами реалізації "хмар".

У приватній хмарі сервіси та інфраструктурні ресурси взаємопов'язані на базі приватної мережі. Така модель гарантує істотно вищий рівень безпеки і контролю, але витрати на програмне і апаратне забезпечення при використанні такого підходу значні в порівнянні з публічними хмарами. Для приватної хмари порівнянно більше можливостей для контролю порівняно з публічною хмарою, просто за рахунок того, що всі компоненти ІТ-інфраструктури залишаються на стороні організації. Таким чином компанії можуть моніторити і управляти хмарним середовищем більш ефективно. •

Високий рівень безпеки забезпечується тим, що споживачем сервісу є одна єдина організація, так що вся інфраструктура може бути оптимально налаштована під існуючі вимоги до захисту даних. Втім, це не означає, що публічні хмари не є безпечними взагалі.

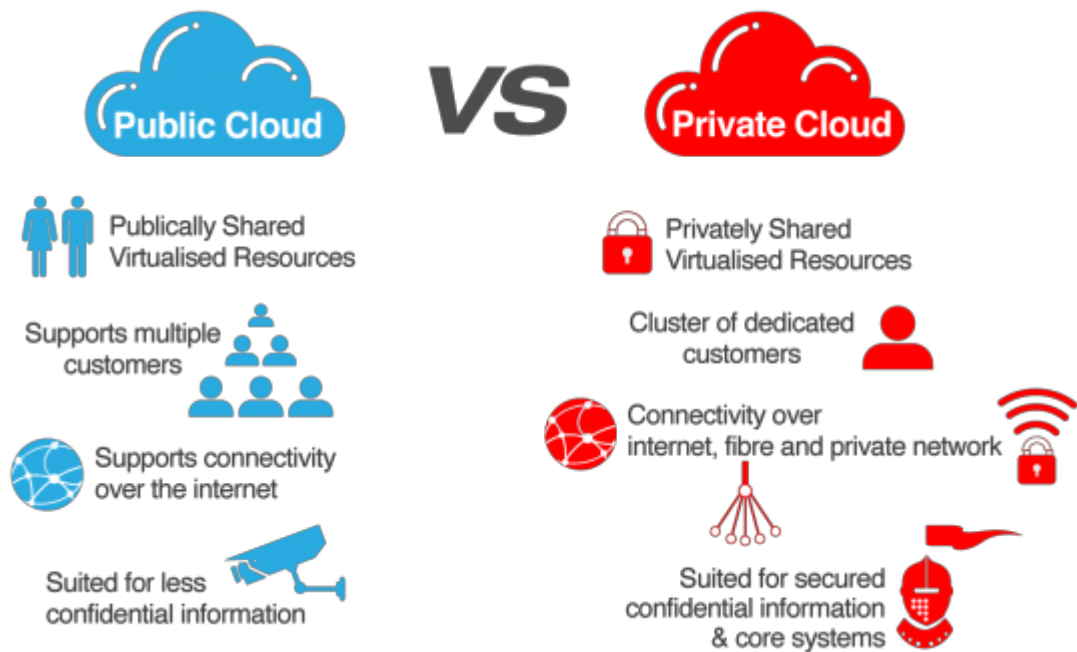


Рисунок 1.1 – Відмінності між різними типами “хмар”[1]

Висока продуктивність приватних хмар пов'язана, зокрема, з тим, що вони оперують в рамках внутрішніх фаєрволів і засобів захисту периметра корпоративної мережі, так що трансфер даних проходить набагато швидше.

При впровадженні приватної хмари підвищується оперативність роботи ІТ-відділу - в будь-який момент за запитом бізнесу він може розгорнути потрібний сервіс за 5-10 хвилин. ІТ-фахівці лише піднімають віртуальну машину з шаблону і встановлюють необхідний сервіс, в той час як компанії з традиційною ІТ-інфраструктурою довелося б замовляти сервер, встановлювати на нього програмне забезпечення і додатки і підключати його до мережі. Одна тільки поставка обладнання часом займає 6-8 тижні [2].

Але слід зазначити, що приватною хмарою необхідно управляти, що тягне за собою витрати на адміністрування і залучення компетентних фахівців у цій галузі. І у порівнянні з публічним хмарою, ризики втрати працездатності сервісів чи втрати даних через фізичних загроз для інфраструктури більш істотні. Мабуть, одним з найбільш істотних недоліків приватного хмари залишається необхідність витрати великих людських і матеріальних ресурсів для його створення і подальшої роботи.

Хмара спільноти, або суспільна хмара. Модель, при якій споживачем є спільнота споживачів з організацій, що мають спільні завдання (наприклад, місії, вимог безпеки, політики і відповідності різним вимогам).

Гібридна хмара - це комбінація з двох або трьох вищеописаних хмар, де різне навантаження може розташовуватися як в приватному, громадському або громадському хмарі. Як правило, гібридна хмара - це більше, ніж просто сума хмар, оскільки йому потрібні механізми та інструменти централізованого управління, розподілу і міграції навантаження між хмарними інфраструктурами.

Цій моделі притаманні п'ять основних характеристик, три сервісні моделі і чотири моделі впровадження. У число характеристик входять: самообслуговування, універсальний доступ по мережі, загальний пул ресурсів, еластичність і облік споживання. Таким чином, сервісні моделі розрізняються за межею контролю споживачем до послуг через надання відповідної інфраструктури.

На рисунку 1.2 в рамках концепції OpenStack наведена приклад моделі системи хмарних обчислень. На цій узагальненій моделі можна відзначити сім елементів: система віртуалізації (гіпервізор), система управління обчислювальними ресурсами (монітор віртуальних машин), сховище даних, сховище образів віртуальних машин, система авторизації і аутентифікації, веб-інтерфейс, та система моніторингу поведінки користувача.

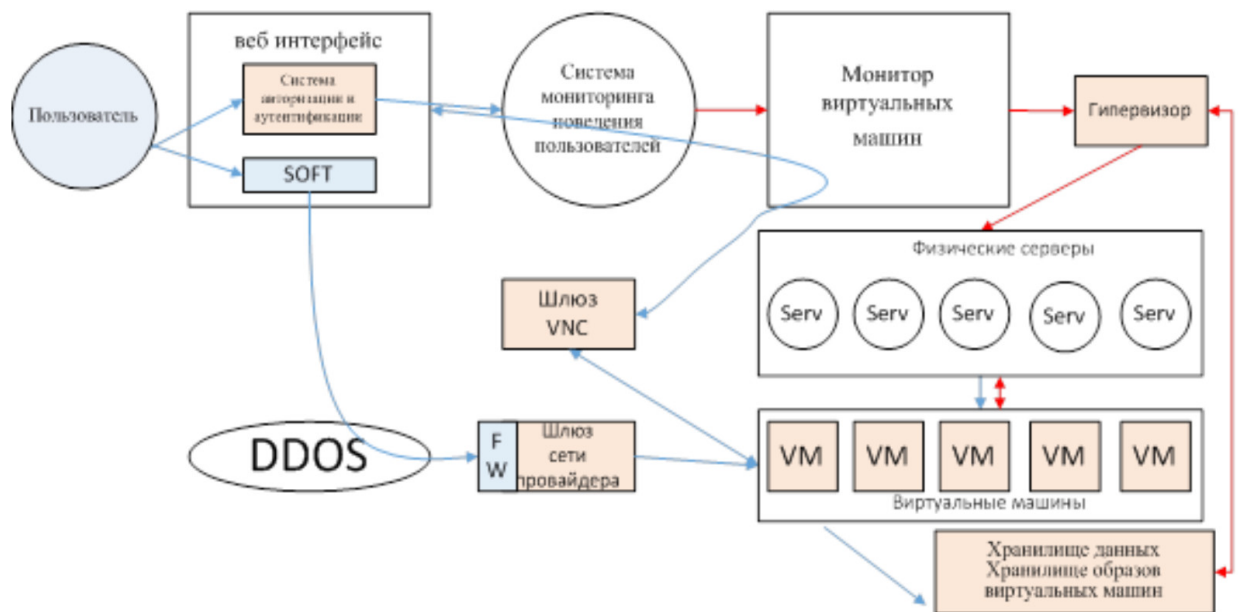


Рисунок 1.2 – Приклад моделі системи “хмарних” обчислень [1]

Система віртуалізації (Гіпервізор) являє собою платформу повної віртуалізації з можливістю використання апаратних можливостей процесора, наприклад серверу кінопоказу.

Система управління обчислювальними ресурсами (монітор віртуальних машин) - це система, яка відповідає за надання користувачам віртуальних машин через управління віртуальним монітором, і так само виділення різного роду ресурсів необхідних користувачеві, зокрема віртуальні мережі, віртуальні сховища даних.

Сховище даних надається користувачеві для зберігання різного роду даних і постійної реплікацією цих даних з використанням віддаленого веб - сервісу. Цей елемент можна використати в якості кіно накопичувача кінотеатру.

Сховище образів віртуальних машин - система зберігання образів віртуальних машин в різних форматах, відстеження доступним образів віртуальних машин для користувача.

Система авторизації і аутентифікації – система, яка використовується всіма суб'єктами для авторизації і аутентифікації системи. Веб-інтерфейс - об'єкт, що надає користувацький інтерфейс.

Система моніторингу поведінки користувача - система, яка побудована на обмеженні і контролі запитів користувачів, а також рекомендацій по виконанню запитів користувачів в системі хмарних обчислень, з можливістю запобігання несанкціонованому доступу до елементів системи хмарних обчислень зокрема до віртуальної машини, гіпервізора, сховища даних тощо.

1.2 Архітектура OpenStack

Проект OpenStack, який також називають “хмарною” операційною системою, складається з ряду окремих проектів, для яких розроблено окремі відкриті підсистеми. Конкретна модифікація OpenStack може включати лише частину з них. Деякі підсистеми можуть використовуватися взагалі автономно, або як частина інших OpenSource проектів. Набір сервісів збільшується від версії до версії проекту OpenStack як за рахунок появи нових, так і за рахунок поділу функціоналу існуючих. Кожен з проектів має свій документований набір REST API, утиліт командного рядка і «рідні» інтерфейси Python, що надають набір функцій, аналогічних утиліт командного рядка [1].

Як і в кожній операційній системі, в OpenStack є користувачі. Після аутентифікації користувач отримує токен для доступу до тих чи інших ресурсів. Користувач створюється не тільки для людини, яка працює в системі, але також і для сервісів. У кожного сервісу є одна або більше точок входу (endpoint). Точка входу є URL, за яким доступний цей сервіс.

Токен – це рядок тексту, що складається з букв і цифр і призначена для доступу до API і ресурсів. Токен видається на обмежений час і при необхідності може бути відкликаний до закінчення терміну дії. Для того щоб користувач отримав токен, він повинен або надати ім'я та пароль, або ім'я і ключ для доступу до API (API key). Токен також містить список ролей, що визначають доступні користувачеві повноваження.

Користувачі та інші ресурси об'єднуються в проекти (в документації англійською мовою також використовується термін *tenant*). Проект є контейнером, який може об'єднувати ресурси окремої організації, що використовує публічну хмару OpenStack, окремого додатка або окремого користувача, – це ви вирішуєте самі. У свою чергу, проекти об'єднуються в домени. Домен - це найбільший контейнер і домени визначають простір імен – область видимості об'єктів. Наприклад, користувачі повинні бути унікальні в рамках одного домена. За замовчуванням Keystone створює домен з ім'ям Default. На рисунку 1.3 наведена базова архітектура платформи OpenStack.

Одним з базових сервісів є OpenStack Compute (Nova). Цей сервіс встановлюється на всіх обчислювальних вузлах кластера. Він надає рівень абстракції віртуального обладнання (процесори, пам'ять, блокові пристрої, мережеві адаптери). Nova забезпечує управління екземплярами віртуальних машин, звертаючись до гіпервізора і віддаючи такі команди, як, наприклад, їх запуск і зупинку. Важливо відзначити, що технології OpenStack незалежні від гіпервізора. Підтримка реалізується через відповідні драйвери в проекті Nova. Розробка і тестування OpenStack ведуться в першу чергу для KVM(Kernel-based Virtual Machine – програмне рішення, що забезпечує віртуалізацію). Більшість впроваджень також зав'язано на гіпервізор KVM.

Крім управління віртуальними машинами, частина сервісів Nova також може забезпечувати управління мережею. Серед розгорнутих в даний час хмар відсоток використання сервісу nova-network досить високий.

Наступний сервіс під назвою OpenStack Networking (Neutron) відповідає за мережеву зв'язаність (рис.1.3). Користувачі можуть самостійно створювати віртуальні мережі, маршрутизатори, призначати IP- адреси. Саме ці функції можна використати при переналаштуванні мережі цифрового кінотеатру, оскільки такі елементи, як цифровий кінопроектор, сервер кінопоказу, ІМВ-блок, ТМС-система між собою організовано у формі інформаційної мережі.

Слід відмітити, що одним з механізмів, що забезпечуються Neutron, називається «плаваючі адреси». Завдяки цьому механізму віртуальні машини можуть отримувати зовнішні фіксовані IP-адреси. Через механізм модулів можна реалізувати такий функціонал, як балансувальник мережевого навантаження як сервіс, брандмауер як сервіс і VPN як сервіс.

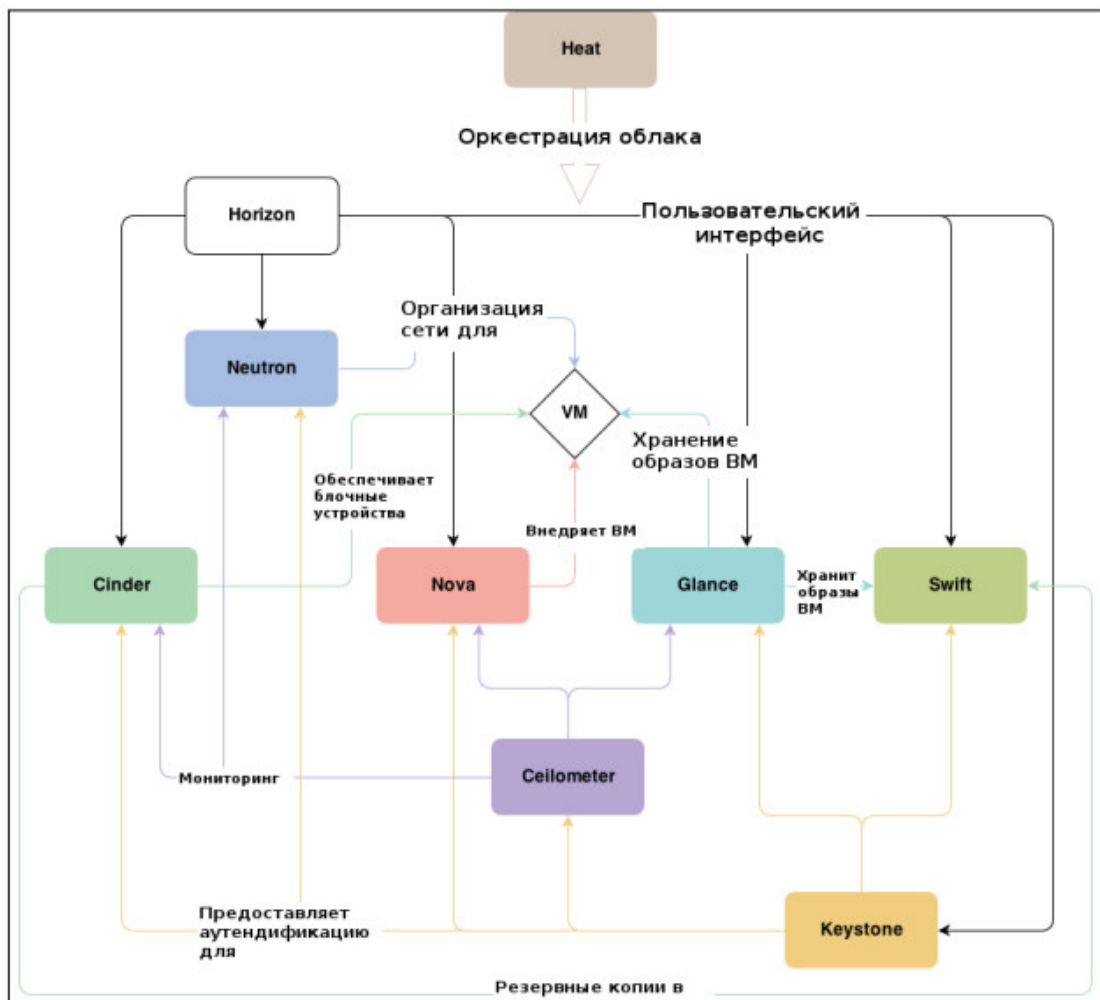


Рисунок 1.3 – Архітектура OpenStack [1]

Служба ідентифікації OpenStack Keystone - це централізований каталог користувачів і сервісів, до яких вони мають доступ. Keystone виступає у вигляді єдиної системи аутентифікації “хмарної” операційної системи. Сервіс підтримує кілька типів аутентифікації, включаючи аутентифікацію по токєну, за допомогою пари ім'я користувача / пароль, і AWS-сумісну аутентифікацію. Keystone підтримує інтеграцію з існуючими сервісами каталогів, наприклад LDAP. Крім цього, Keystone є каталогом служб, доступних в OpenStack,

підтримуючи довідник реквізитів для звернення до API відповідних сервісів, а також реалізує політику рольового контролю доступу. Keystone перевіряє дійсність облікових записів користувачів, перевіряє зіставлення користувачів проектам OpenStack і ролям і в разі успіху видає токен на доступ до інших сервісів. Також Keystone веде каталог служб. Цей елемент можна використати при плануванні плей-листів кінопоказу а такою при формуванні бази даних доступу до системи цифрового кінопоказу.

OpenStack Image Service (Glance) веде каталог образів віртуальних машин, які користувачі можуть застосовувати як шаблони для запуску примірників віртуальних машин в хмарі. Також даний сервіс надає функціонал резервного копіювання та створення моментальних знімків. Glance підтримує безліч різних форматів, включаючи vhd, vmdk, vdi, iso, qcow2, ami і ін.

Сервіс OpenStack Block Storage (Cinder) управляє блоковим сховищем, яке можуть використовувати запущені екземпляри віртуальних машин. Це постійне сховище інформації для віртуальних машин. Можна використовувати моментальні знімки для збереження і відновлення інформації або для клонування. Найчастіше з cinder використовують сховище на основі Linux-серверів, проте є і модулі для апаратних сховищ.

Сервіс OpenStack Telemetry (Ceilometer) являє собою централізоване джерело інформації по хмарам і даними моніторингу. Цей компонент забезпечує можливість білінгу для OpenStack.

OpenStack Orchestration (Heat) - сервіс, завдання якого – забезпечення життєвого циклу додатків в хмарній інфраструктурі. За допомогою шаблону формату AWS CloudFormation сервіс управляє іншими сервісами OpenStack, дозволяючи створити більшість типів ресурсів (віртуальні машини, томів, плаваючі IP, користувачі, групи безпеки і т.д.). Heat за допомогою даних Ceilometer також може здійснювати автоматичне масштабування додатків. Шаблони описують відносини між ресурсами, і це дозволяє сервісу Heat

здійснювати виклики API OpenStack в правильному порядку, наприклад спочатку створити сервер, а потім підключити до нього декілька томів.

І нарешті, найближчий до користувача хмари сервіс OpenStack Dashboard (Horizon), що дозволяє управляти ресурсами хмари через веб-консоль.

1.2.1 Сервіс Swift як образ кінонакопичувача

Сервіс OpenStack Object Storage (Swift), крім Nova, є одним з двох перших проєктів, що з'явилися в OpenStack. Спочатку він називався Rackspace Cloud Files. Сервіс являє собою об'єктне сховище, що дозволяє користувачам зберігати файли. В якості останніх в мережі цифрового кінотеатру можуть виступати мастер-копії кінофільмів.

Swift має розподілену архітектуру, забезпечуючи горизонтальне масштабування, а також надмірність і реплікацію для відмовостійкості. Swift орієнтований переважно на статичні дані, такі як образи віртуальних машин, резервні копії і архіви. Swift – це програмно визначене сховище (Software-defined storage, SDS), що працює з об'єктами. Об'єктне сховище, на відміну від файлового або блочного, надає доступ не до файлів і блоковим пристроям, а до об'єктів в єдиному просторі імен. У об'єктного сховища є свій API, і зазвичай доступ до об'єктів здійснюється по протоколу HTTP. Таке сховище абстрагує об'єкти від їх матеріального становища і дозволяє здійснювати масштабування незалежно від лежачої під сховищем фізичної інфраструктури. Також перевагою об'єктного сховища є можливість розподіляти запити по великому числу серверів, що зберігають дані.

Основні властивості архітектури Swift наступні: лінійна масштабованість; відсутність ексклюзивних ролей або майстрів операцій; механізми реплікації і самовідновлення. За замовчуванням у кожного об'єкта три репліки. Відповідно, не потрібен RAID-масив на вузлах зберігання мережі; підтримка великих об'єктів (за замовчуванням до п'яти гігабайт).

Архітектура Swift забезпечує «консистентність в кінцевому рахунку»(англомовний термін – eventually consistent). Цей термін означає, що під час відсутності змін даних в кінцевому рахунку всі запити будуть повертати останнім оновлене значення. Тобто, коли ви завантажуєте об'єкт в кластер, ви отримаєте підтвердження, що об'єкт записаний, коли прийдуть підтвердження від двох вузлів. Решта вузлів отримають об'єкт «в кінці-кінців».

Фізична структура Swift не перетинається з логічною структурою і складається з наступних компонентів:

- регіон – відповідає одному майданчику або центру обробки даних. Якщо кластер містить кілька регіонів, то Swift намагатиметься прочитати дані з найближчого, засновуючи своє рішення на затримці при передачі даних. Під час запису за замовчуванням Swift пише інформацію в усі регіони одночасно. Кожен регіон має свою точку входу API;

- зона – це набір серверів в регіоні, що характеризується загальним компонентом доступності, наприклад набір стійок, зав'язаних на одну лінію живлення, сервери, підключені до однієї пари комутаторів, леза в одному кошику і т. п. Як правило, в кластері - кілька зон. Документація OpenStack рекомендує створювати мінімум п'ять зон;

- сервери – окремі сервери, що зберігають дані;
- диски – диски серверів зберігання.

Об'єкти зберігаються як файли на дисках з файловою системою, що дозволяє зберігати розширені атрибути файлів. Рекомендуються ext4 і XFS. Також рекомендується досить великий розмір inode (індексного дискриптору) для цілей зберігання метаданих. Наприклад, можна задати розмір, починаючи від 1024.

1.2.2 Організація вузлів зберігання сервісу Swift

Для того, щоб організувати віртуалізацію кінонакопичувача в мережі цифрового кінотеатру формату D-Cinema необхідно виконати певні процедурні налаштування сервісу Swift [1]. Ці налаштування починаються з того, що спочатку слід встановити необхідні пакети на дві віртуальні машини sw2.test.local і sw3.test.local. Команди потрібно виконувати на обох серверах:

```
[Root @ sw2 i 3 ~] # yum install openstack-swift-account openstack-swift-container openstack-swift-object.
```

Кожній з двох віртуальних машин додамо по два локальних диска і на кожному з них створимо по одному розділу.

Імена блокових пристроїв залежать від системи віртуалізації і типу емульованого пристрою. У разі KVM VirtIO це / dev / vdb і / dev / vdc. Створимо на обох розділах віртуальних машин sw2 і sw3 файлову систему XFS:

```
[Root @ sw2 i 3 ~] # mkfs.xfs / dev / vdb1
```

```
[Root @ sw2 i 3 ~] # mkfs.xfs / dev / vdc1
```

Файлові системи необхідно змонтувати в піддиректорії

/ Srv / node:

```
[Root @ sw2 i 3 ~] # mkdir -p / srv / node / vd {b, c} 1
```

```
[Root @ sw2 i 3 ~] # chown -R swift: swift / srv / node /
```

```
[Root @ sw2 i 3 ~] # echo "/ dev / vdb1 / srv / node / vdb1 xfs defaults 0 0"
```

```
>> / etc /
```

stab

```
[Root @ sw2 i 3 ~] # echo "/ dev / vdc1 / srv / node / vdc1 xfs defaults 0 0"
```

```
>> / etc /
```

fstab

```
[Root @ sw2 i 3 ~] # mount -a
```

Для того щоб Swift виконував реплікацію, необхідно налаштувати демон rsyncd. Створюємо на обох вузлах файл /etc/rsyncd.conf такого змісту:

```
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = IP-адреса вузла

[Account]
max connections = 25
path = / srv / node /
read only = false
lock file = /var/lock/account.lock

[Container]
max connections = 25
path = / srv / node /
read only = false
lock file = /var/lock/container.lock

[Object]
max connections = 25
path = / srv / node /
read only = false
lock file = /var/lock/object.lock
```

Ознайомитись зі значенням параметрів конфігураційного файлу можна на man-сторінці rsyncd.conf (5). Запускаємо і активуємо сервіс:

```
[Root @ sw2 i 3 ~] # systemctl enable rsyncd.service
```

```
[Root @ sw2 i 3 ~] # systemctl start rsyncd.service
```

Перевіряємо роботу демона rsync для обох серверів:

```
[Root @ sw2 i 3 ~] # rsync IP-адреса ::
```

```
[Root @ sw2 i 3 ~] # rsync IP-адреса :: object
```

```
[Root @ sw2 i 3 ~] # rsync IP-адреса :: account
```

```
[Root @ sw2 i 3 ~] # rsync IP-адреса :: container
```

Тепер на кожному з двох вузлів необхідно створити по три конфігураційних файли сервісів. Для управління конфігурацією сервісів Swift використовуємо систему Paste.deploy ([Http://pythonpaste.org/deploy/](http://pythonpaste.org/deploy/)). Значення опцій описано на сторінці

http://docs.openstack.org/developer/swift/deployment_guide.html.

Конфігураційний файл /etc/swift/account-server.conf:

[DEFAULT]

bind_ip = IP-адреса сервера

bind_port = 6202

user = swift

workers = 2

swift_dir = / etc / swift

devices = / srv / node

[Pipeline: main]

pipeline = healthcheck recon account-server

[App: account-server]

use = egg: swift # account

[Filter: recon]

use = egg: swift # recon

recon_cache_path = / var / cache / swift

[Filter: healthcheck]

use = egg: swift # healthcheck

[Account-replicator]

[Account-auditor]

[Account-reaper]

Конфігураційний файл /etc/swift/object-server.conf:

[DEFAULT]

bind_ip = IP-адреса сервера

bind_port = 6200

user = swift


```
swift_dir = / etc / swift
devices = / srv / node
workers = 3
[Pipeline: main]
pipeline = healthcheck recon object-server
[Filter: recon]
use = egg: swift # recon
recon_cache_path = / var / cache / swift
[App: object-server]
use = egg: swift # object
[Filter: healthcheck]
use = egg: swift # healthcheck
[Object-replicator]
[Object-updater]
[Object-auditor]
Конфігураційний файл /etc/swift/container-server.conf:
[DEFAULT]
bind_ip = IP-адреса сервера
bind_port = 6201
ser = swift
swift_dir = / etc / swift
devices = / srv / node
workers = 2
[Pipeline: main]
pipeline = healthcheck recon container-server
[Filter: recon]
use = egg: swift # recon
recon_cache_path = / var / cache / swift
[App: container-server]
use = egg: swift # container
```

```
[Filter: healthcheck]1  
use = egg: swift # healthcheck  
[Container-replicator]  
[Container-updater]  
[Container-auditor]  
[Container-sync].
```

1.3 Захист даних в “хмарних” системах

Хоча хмарні обчислення сьогодні вже не є відносно новою технологією, питання забезпечення їх інформаційної безпеки продовжують залишатися слабким місцем. У зв'язку з технологічними особливостями, використовуваними для побудови структури хмарних обчислень, до стандартних типів загроз, які є наслідком розміщення ресурсів на фізичних серверах, додалися складності, пов'язані з контролем хмарного середовища віртуалізації, трафіку між гостьовими машинами та розмежуванням прав доступу. Більш того, розподілена і відкрита структура хмарних обчислень з мультидоменною і розрахованою на багато користувачів структурою стала дуже привабливою мішенню для потенційних зловмисників. Як вже було відмічено у попередньому пункту, архітектура хмарних сервісів складається з трьох взаємозалежних рівнів: інфраструктура, платформа і додатки. І кожен з цих рівнів може бути уразливий до програмних і конфігураційних помилок, допущених користувачами або провайдерами сервісу. Система хмарних обчислень може піддаватися декільком видам загроз безпеки - включаючи погрози цілісності, конфіденційності та доступності її ресурсів, даних і віртуальної інфраструктури, які можуть бути використані нецільовим чином, наприклад, в якості майданчика для поширення нових атак [3].

Одним з основних моментів, який необхідно враховувати стосовно безпеки в “хмарі”, полягає у тому, що відповідальність за використання ресурсів поділяється між клієнтом і постачальником “хмарного” сервісу. І

необхідно розуміти при проектуванні мережі, де закінчується відповідальність провайдера хмарних обчислень і починається відповідальність клієнта. При побудові складних систем (однієї з різновидів яких є хмари) застосовують архітектурну концепцію багаторівневої безпеки (Defensein-Depth) – механізм, який використовує кілька рівнів захисту, для того щоб збільшити час, який витрачається на спроби зламу системи; а також вести підрахунок кількості спроб злому для прийняття рішення про блокування атаки.

Відповідно, при побудові системи безпеки середовища хмар також можна виділити свої шари контролю і доступу. Хмара комбінує можливості користувача і постачальника, брандмауери і різновиди способів ізоляції. При цьому окремі елементи безпеки можуть контролюватися користувачем незалежно від провайдера.

У моделі IaaS на стороні замовника можна побудувати свої власні технічні засоби забезпечення безпеки. Клієнт може мати повний контроль над реальною конфігурацією сервера, що гарантує йому більший контроль ризиків безпеки оточення і даних.

У PaaS постачальник управляє лише апаратною платформою і операційною системою, що обмежує можливості підприємства замовника в управлінні ризиками на цих рівнях.

У моделі SaaS як платформа, так і інфраструктура повністю управляється провайдером хмарних послуг. Це означає, що, якщо операційна система або сервіс не налаштовані належним чином, то дані на більш високому прикладному рівні можуть бути в небезпеці. Для покупців у цьому випадку необов'язково знати, як надаються ці послуги (які включають в себе мережу, сервери, операційні системи, сховища і навіть окремі функції додатків). Користувачу важливо, щоб сервіс був досить дешевий і доступний в будь-який час, коли він необхідний. Тому багато деталей функціонування сервісу і його інфраструктура виявляються прихованими для користувача. У можливостях управління клієнт виявляється обмеженим тільки мінімальним

набором налаштувань конфігурації програми під свої потреби. Відповідальність постачальника хмарного сервісу починається з фізичної безпеки і безпеки середовища. Цей рівень безпеки - високорівневий, так як він пов'язаний з керованістю хмарою як єдиною інформаційною системою. Саме постачальник хмарного сервісу здійснює експлуатацію фізичних серверів центрів обробки даних, тому клієнт так само, як і у випадку зі звичайним ЦОД, повинен розглянути наступні ключові моменти: фізичний доступ персоналу до серверів і мережевої інфраструктури, засоби пожежної сигналізації та пожежогасіння, кліматичний і температурний контроль над серверами і іншими апаратними засобами, знищення виведених з експлуатації пристроїв зберігання даних. На відміну від фізичної безпеки, мережева безпека в першу чергу є побудова надійної моделі загроз, що включає в себе захист від вторгнень і міжмережевий екран. Під використанням міжмережевого екрану мається на увазі роботу фільтра з метою розмежувати внутрішні мережі ЦОД на підмережі з різним рівнем довіри. Це можуть бути окремо сервери, доступні з Інтернету, або сервери з внутрішніх мереж. Доступ через Інтернет до управління обчислювальною потужністю хмари - одна з ключових характеристик хмарних обчислень. У більшості традиційних ЦОД доступ інженерів до серверів контролюється на фізичному рівні, в хмарних середовищах вони працюють через Інтернет. Розмежування контролю доступу та забезпечення прозорості змін на системному рівні є одними з головних критеріїв захисту.

Прийнято вважати, що приватні хмари є найбільш безпечними, оскільки вони дозволяють впровадити власні кошти шифрування і захисту ще на етапі їх створення, а також через те, що дані залишаються в існуючій інфраструктурі компанії. Однак, якщо дані не захищені належним чином в хмарі вони можуть бути втрачені або пошкоджені незалежно від того приватна ця хмара або публічна.

Цікава наявність неконтрольованої системами безпеки сліпої зони - трафіку між віртуальними серверами в хмарі. Традиційні засоби моніторингу

працюють з використанням віддзеркалення трафіку з портів мережевих пристроїв і сенсорів, які здатні захоплювати і аналізувати цей трафік. Однак канали передачі даних між ВМ створюються в гіпервізорі. Шкідливий трафік і дані можуть переміщатися між ВМ без виходу в реальну мережу, що означає, що атака буде не помічена традиційними інструментами. Дані, що зберігаються на виключених ВМ, також є уразливими, у випадках, коли в основній ОС, на якій вони розміщуються, контроль доступу не налаштований належним чином, або не встановлені оновлення, що виправляють критичні уразливості. На іншій стороні (в сторону зменшення безпеки) прийнято розташовувати публічні хмари [4].

Існують і зовнішні загрози безпеки, такі як, наприклад, віддалені хакерські атаки. У публічних хмарах розміщується величезна кількість корпоративних даних, що робить їх привабливими для злоумисників.

Аналогічно атака на мережеве сховище резервних копій великої кількості компаній може дати більше даних, ніж злом сховища, що належить тільки одній організації. І навіть коли сховище даних досить добре захищене від зовнішніх атак, а контроль і розмежування доступу надає тільки мінімальні повноваження особливо довіреним особам, все ще залишаються відкритими питання безпеки при передачі даних між клієнтом і хмарною інфраструктурою.

Існує кілька способів захистити дані в хмарі. Частина з них вже згадувалися - це контроль доступу і моніторинг. Однак найбільш ефективним і при цьому універсальним способом забезпечити захист даних, їх конфіденційність і цілісність - це використання шифрування даних при їх передачі по інформаційним мережам і при зберіганні всередині хмари. Наприклад, в керівництві по інформаційній безпеці, розробленому Альянсом безпеки хмар, стверджується, що шифрування надає переваги найменшій залежності як від провайдера хмарного сервісу, так і від експлуатаційних помилок. Захист даних, заснований на шифруванні, робить ці дані марними для будь-якої особи, що не має ключів для їх дешифрування. І не важливо,

знаходяться ці дані в процесі передачі або зберігання, вони залишаються захищеними. Власник ключів шифрування підтримує безпеку даних, і приймає рішення кому, і до яких даними надавати доступ. Процедура шифрування може бути вбудована в існуючий робочий процес хмарних сервісів. Наприклад, адміністратор може зашифровувати всі дані резервного копіювання перед відправкою їх в хмарне сховище. Співробітник організації може захистити корпоративну інтелектуальну власність, перш ніж покласти його в приватну хмару. Представник компанії може зашифрувати особисті контракти клієнтів, перш ніж відправити їх в спільне робоче місце в публічну хмару.

Висновки

В першому розділі дослідження розглянуто особливості побудови архітектури OpenStack, проаналізовано основні рівні “хмарної” платформи. Зокрема, за результатами розгляду знайдено, що деякі елементи платформи можна використати як захищені елементи мережі цифрового кінотеатру формату D-Cinema. Наприклад, за набором функцій сервіс Swift платформи OpenStack можна використати, як елемент кінонакопичувача типової схеми цифрового кінотеатру. Окремо, розглянуто сильні та слабкі сторони різних тип “хмар” та визначено положення, які слід враховувати при проведенні етапів аутентифікації користувачів системи даних, яку може відігравати структура сучасного багатозального цифрового кінотеатру.

2 ОСОБЛИВОСТІ СТВОРЕННЯ МЕРЕЖІ DIGITAL CINEMA

2.1 Особливості створення мережі кінотеатрів D-cinema

На сьогодні система цифрового кінотеатру Digital Cinema - гнучка система, яка дозволяє глядачеві залу отримати найкраще враження від перегляду. Система повинна включати відео та аудіо компоненти і відповідати за принципами функціонування відкритим стандартам, встановленим DCI і SMPTE. Поруч з цим, система повинна бути ще надзвичайно надійною, універсальною, та мати властивість простої модернізації. Основними функціями сучасної мережі кінотеатрів можна вважати наявність безперервного показу цифрових кінофільмів а також можливість відтворення у глядацькій залі різного роду захищеного аудіовізуального контенту. Для забезпечення цього, у структурі Digital Cinema можна виділити наступні складові:

- сервер кінопоказу, який об'єднує функції потокового сервера і декодера цифрового контенту;
- сервер зберігання даних з великим об'ємом пам'яті (кінонакопичувач для мультіплексних кінотеатрів);
- звуковий кінопроцесор з адаптером для відтворення різних звукових форматів;
- програмне забезпечення, яке відображає роботу системи управління кінотеатром Theatre Management System (TMS);
- комутатори 1000Base-T Ethernet, для забезпечення комутації цифрових потоків багатозального глядацького комплексу кіно.
- USB клавіатура, USB миша і комп'ютерний монітор;
- цифровий лазерний кінопроектор.

Додатково в систему може входити мережевий інтерфейс системи автоматизації кінотеатру. На рисунку 2.1 наведена типова схема мережі Digital Cinema нижнього рівня, і тут для зручності показано окремо наявні

Ethernet-з'єднання (штрих-пунктирні лінії зв'язку). Наведена схема відображає функціональні зв'язки для обладнання однієї глядацької зали.

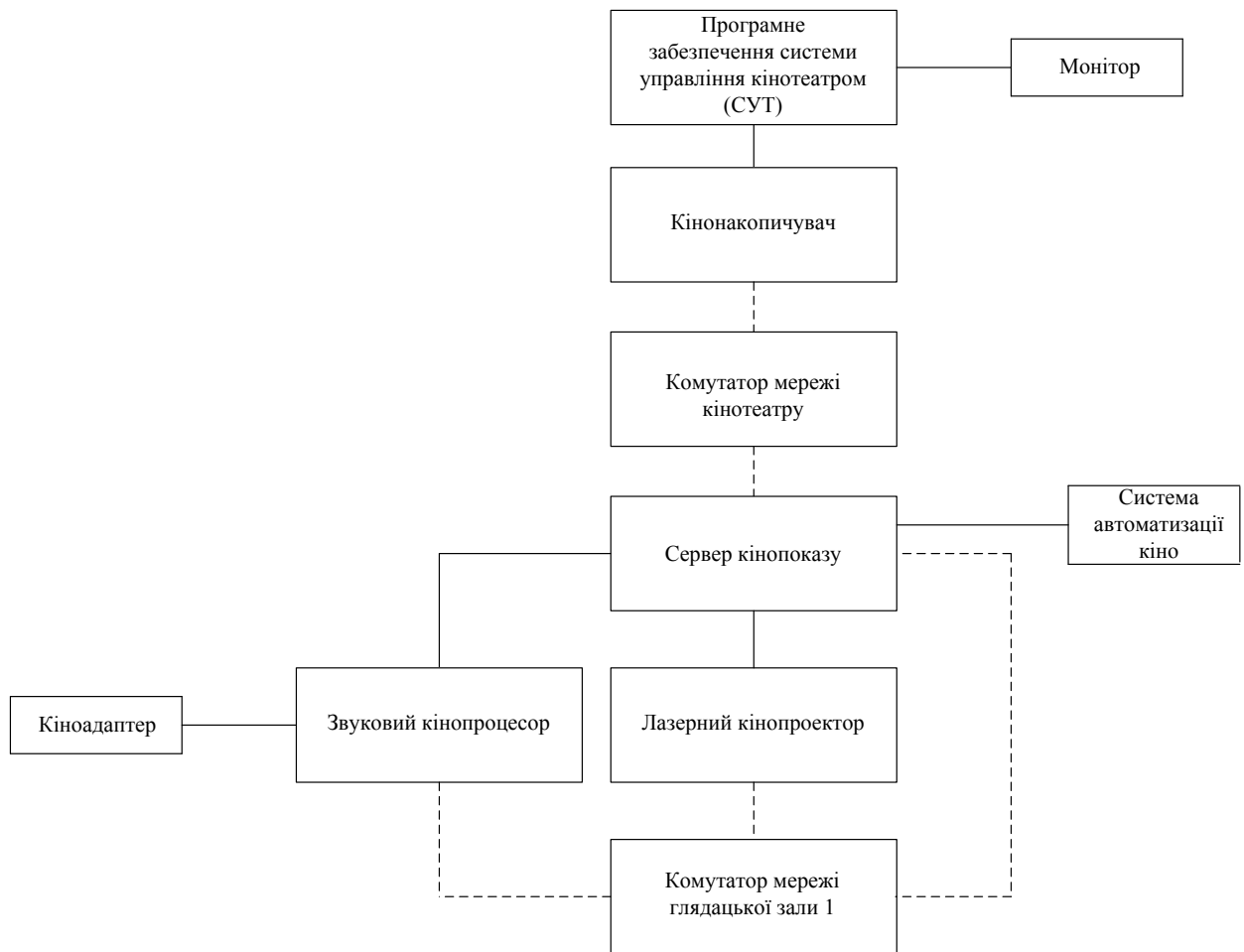


Рисунок 2.1 – Типова схема мережі Digital Cinema

Система автоматизації кіно забезпечує управління в глядацькій залі освітленням, завісою і в деяких випадках може мати функції управління лазерним кінопроектором в кіноапаратній. Крім цього, ця система забезпечує завантаження та редагування плейліста сеансу, на її основі можна запрограмувати початок, зупинку, паузу кінопоказу.

Сервер кінопоказу об'єднує функції потокового серверу та декодери цифрового контенту та може використовуватись при складанні листів розкладу фільмів в кінотеатрі. Цей елемент мережі отримує зашифрований і закодований контент від кінонакопичувача. Причому, сам цифровий кіно пакет, який визначає кінокопію, містить окремі mxf-файли для аудіо та відео

потоків. Далі ці файли через шифрований канал передаються на кінопроектор обраної глядацької зали де відбувається їх дешифрування та проекція. Крім цього, сервер кінопоказу передає аудіо дані цифрового потоку на звуковий кіно процесор мережі для подальшого відтворення через звукові підсилювачі на основі акустичної багатоканальної системи в залі (на рисунку 3.1 не показано).

Слід відмітити, що один сервер кінопоказу рекомендується використовувати для обслуговування не більше 3 глядацьких залів мультиплексу. Для підключення серверу до мережі залів використовується комутатор зали та відповідно Ethernet-з'єднання. Натомість для підключення до системи автоматизації кіно залучається послідовний порт RS-232 серверу, який на основі відповідної лінії зв'язку підключається до програмованого контролера, який підключено до системи автоматизації кінотеатру.

Кінонакопичувач є центральним елементом мережі кінотеатру і використовується для зберігання кінокопій, які на даний момент відтворюються в глядацьких залах мережі. Слід відмітити, що даний елемент може використовуватись і для виконання функцій сервера системи управління кінотеатром і він не підключається безпосередньо до однієї з мережі глядацької зали. Ця обставина дозволяє використати цей елемент для проведення процедури віртуалізації сховища зберігання контенту шляхом впровадження елементів “хмарної” платформи Open Stack.

Клієнтський компонент системи управління кінотеатром забезпечує в мережі Digital Cinema контроль та управління усім комплексом кінотеатру. Про особливості цього елемента буде наведено нижче в магістерському дослідженні.

Цифровий лазерний кінопроектор в мережі D-Cinema підключається до комутатора кінозали, адже через макроси є можливість передавати до кінопроектору команди, наприклад, з управління джерелом світла проектора або забезпечувати регулювання дімером освітлення. Система Digital Cinema підтримує Ethernet команди (лампа, диммер, макроси), адресовані цифровому

кінопроектору. Щоб задати IP адресу проектора, потрібно використовувати програмне забезпечення проектора. Основна адреса цифрового кінопроектора - це адреса головки TI™ DLP™, яка використовується при передачі субтитрів і ключів каналного шифрування. Однак більшість проекторів мають також блок контролера з власною IP адресою. Цей блок служить для обробки команд автоматичного управління проектором, і його IP адресу потрібно правильно задати.

Звуковий кінопроцесор використовується в схемі (рис.3.1) для відтворення аудіо даних різних форматів та з різних звукових джерел (не особливо з цифрового кінопаketу).

На рисунку 2.2 наведено верхній рівень взаємодії елементів мережі кінотеатру. Тут слід відзначити наступні технічні особливості:

- сервер кінопоказу забезпечує роботу певної глядацької зали зі своїм кінопроектором та кінопроцесором;
- схема дозволяє підключати контент, який передається через супутникові лінії зв'язку;
- схема може бути розширена через використання порту Theatre Network;
- схема має високий рівень гнучкості, адже комутатор є універсальним і має можливість зовнішнього програмування фізичних портів.

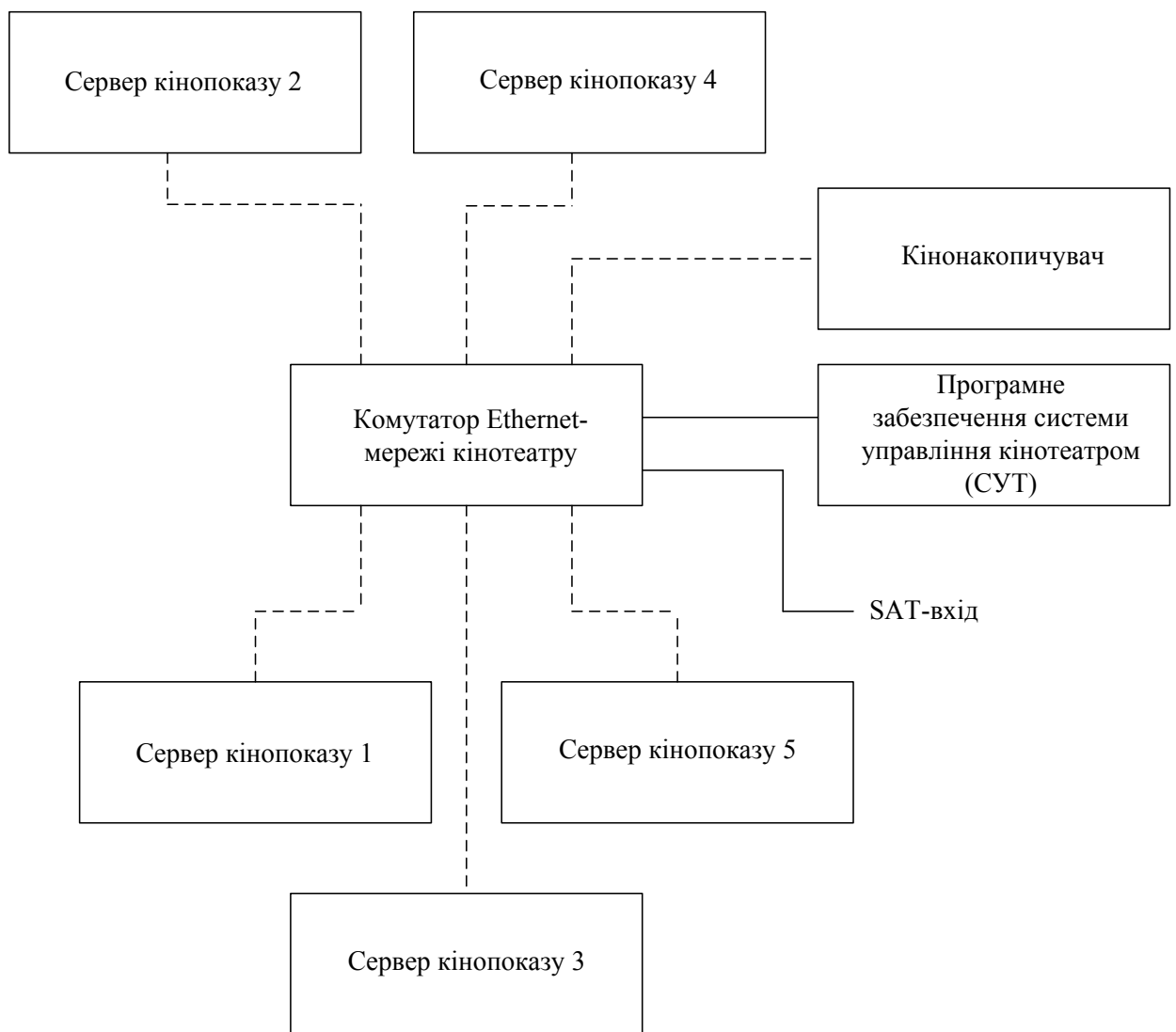


Рисунок 2.2 – Типова схема верхнього рівня мережі Digital Cinema

Аналізуючи схему на рисунку 2.2 можна відмітити, що в мультиплексних кінотеатрах можна встановити систему Digital Cinema в декількох залах і створити мережу в кінотеатрі, щоб керувати кожним з цих залів засобами системи управління кінотеатром (рис.2.1). Тобто, перш за все треба встановити системи у кожному з кінозалів, і після цього потрібно підключити сервер кінопоказу, а також кіно накопичувач для кожного залу до комутатору мережі кінотеатра.

Сервер кінозалу і кіносервер для зберігання даних (кінонакопичувач) - це основні компоненти системи цифрового кінотеатру. Сервер зчитує файли аудіо і відео даних, дешифрує і декодує їх і потім знову шифрує відеодані для забезпечення захисту відеопотоку. Далі сервер передає відеодані по

шифрованому каналу на цифровий кінопроектор місцевого кінозалу для дешифрування і відтворення. Крім цього сервер передає аудіо складову потоку даних або на кіноадаптер і через нього на кінопроцесор іншої фірми, або безпосередньо на кінопроцесор для відтворення звуку за допомогою встановлених в цьому залі підсилювачів і акустичних систем. Сервер кінопоказу дозволяє централізовано зберігати весь контент, використовуючи змінні жорсткі диски, накопичувачі USB 3.0, DVD або FTP-сервери. Цей апарат розподіляє файли по всім системам Digital Cinema мультиплексного кінотеатру. Для показу контенту в певному залі сервер з'єднується з мережею залу через Ethernet-комутатор. Цей комутатор підключається до старих моделей кінопроцесорів. Він забезпечує з'єднання з кінопроцесором і з цифровим кінопроектором залу. Крім того, його можна підключити до інтерфейсу системи автоматизації кінотеатру, який використовує порт RS-232 з кодом ASCII. В цьому випадку сервер буде керувати обладнанням залу, підключеним до мережі Ethernet (завіса, освітлення, тощо.).

Кінонакопичувач об'єднує всі кінозали через комутатор 1000Base-T Ethernet. У кінотеатрах з декількома залами всі кінонакопичувачу (по одному на зал) також підключені до цього комутатора. Підключення до мережі кінотеатру дозволяє передавати на сервер кожного залу програму показу, призначену для цього залу. Програма показу складається з послідовності роликів контенту і команд управління обладнанням кінозалу. Програмне забезпечення системи управління кінотеатром здійснює контроль і управління всією мережею кінотеатру. Центральною прикладною програмою є TMS-сервер. Цей сервер може виконуватись на основі кіно накопичувача (або на відповідно прошивці серверу кінопоказу, якщо в кінотеатрі не більше трьох залів).

TMS клієнт підтримує інтерфейс користувача для управління системою Digital Cinema. Програма-клієнт виконується на кіно сервері зберігання даних, на будь-якому мережевому сервері кінопоказу або на віддаленому ПК. Це дозволяє зареєстрованим користувачам дистанційно (наприклад, з

кабінету керуючого) стежити за станом кожного залу, управляти відтворенням, створювати програми показу і складати розклад. Окремо, лід відмітити, що система управління кінотеатром дозволяє провести при діагностиці пошук несправностей обладнання та виконати контроль стану всього цифрового обладнання мережі Digital Cinema.

2.2 Система управління кінотеатром

Система управління кінотеатром для короткострокового показу, централізовано використовує всі процеси та процедури, які задіяні в цифровому кінотеатрі. Це дає змогу повністю зосередитися на розвитку індустрії та економії на інших технологіях. Система управління кінотеатром - це мережева система кінематографії для централізованого управління та експлуатації обладнання, такого як кінопроектори та сервери в кількох кінопроекційних кімнатах. Без розгортання з одного місця в кінотеатрі ви можете завантажувати нові цифрові кінопакети, вводити ключі захисту, збирати, редагувати та планувати покази та автоматично поширювати їх на потрібні глядацькі зали мультиплексу.

Фактично система управління кінотеатром складається з деяких основних функцій та медіа-бібліотеки, яка є центральним сховищем даних для цифрових фільмів (рисунок 2.3). Специфікації системи цифрового кінематографу охоплюють не тільки системи керування екраном для локальної роботи, вони також включають положення та рекомендації для мережевих систем управління.

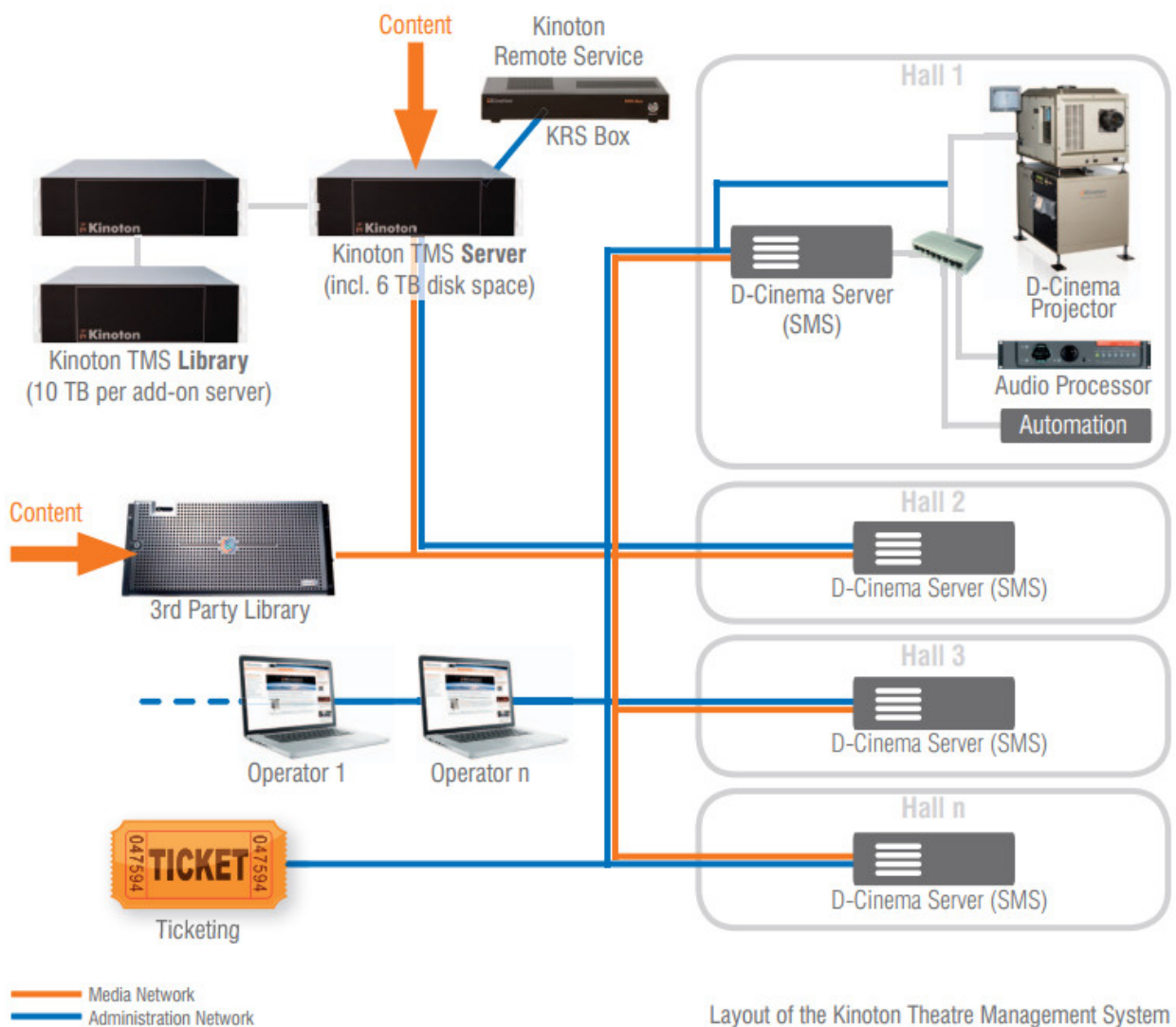


Рисунок 2.3 – Система управління кінотеатром фірми Kinoton [5]

Коли нові кінокопії та ключі надходять до кінотеатру, вони завантажуються в локальний дисковий масив сервера, перевіряються та копіюються в центральну базу даних. Інтерфейс користувача системи управління кінотеатром дозволяє отримати доступ до усіх проекторів, їх списків відтворення та локально доступних макросів автоматизації для всіх підключених серверів D-Cinema, а також для всього вмісту, що зберігається в центральній бібліотеці кінонакопичувача. У результаті під час складання нового показу можна вільно вибирати з усього вмісту контенту, доступного в кінотеатрі, не турбуючись про те, де він фізично розташований. Саме цей аспект можна використати при проведенні віртуалізації ресурсів кінотеатру

під час запровадження “хмарної” моделі сховища на основі платформи Open Stack.

Також можна вручну зберегти та запустити список відтворення до зали або керувати ним за допомогою програмного планувальника. Система управління кінотеатром автоматично передає новий контент разом з усіма необхідними файлами, ключами тощо на вибраний сервер екрана (рис.2.4).

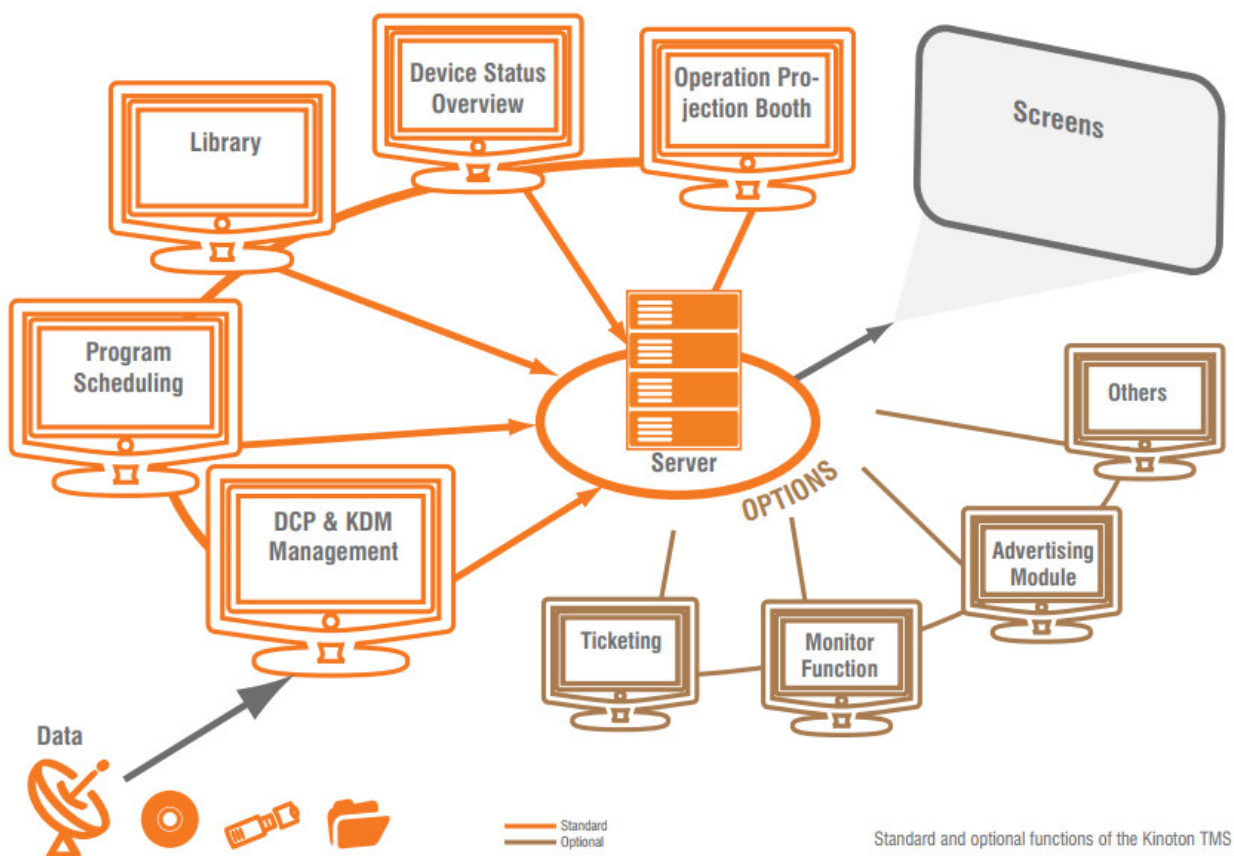


Рисунок 2.4 – Схематичне зображення функцій системи управління кінотеатром фірми Kinoton [5]

Система робить це розумно: щоб не перевантажувати загальну мережу або її окремі сервери, що може перешкоджати демонстраціям, він завантажує вміст з різних джерел, залежно від наявності. Після цього можна встановити час демонстрації або з користувальницького інтерфейсу відповідного сервера D-Cinema або безпосередньо через його планувальник, налаштувавши тим самим систему автоматизації кінотеатру або центральний контролер зали - з будь-якої точки в мережі.

Зокрема, в мультиплексних кінотеатрах раніше централізована підготовка плейлистів, керування вмістом та планування роботи мережі займала величезну кількість часу. Перед появою систем управління кінотеатром кожного разу, коли приходить жорсткий диск з кінофільмом, треба було ходити з одного сервера D-Cinema на інший, і кожного разу повторюючи тривалий процес надходження нових фільмів на сервер, введення ключів, і зібрання списків відтворення, пов'язаних із глядацькою залю. Завдяки системі управління кінотеатром, це просто можна зробити з одного робочого місця, а також швидко змінювати зали показу за короткий час при натисканні кнопки - за умови, звичайно, що ліцензійний ключ фільму дійсний для обох випадків. У той же час користувальницький інтерфейс надає інформацію про поточний статус системи всіх підключених пристроїв у всіх глядацьких залах.

Висновки

В другому розділі дослідження проаналізовано основні елементи, які складають сьогодні сучасну мережу кінотеатрів Digital Cinema. Зокрема, вказано, що окремі елементи за рахунок їх основних функцій можна об'єднати між собою, а такі елементи як, кінонакопичувачі з урахуванням дотримання відповідного захисту даних, можна віртуалізувати шляхом переведення останніх у “хмарне” середовище. Окремо наведено функції системи управління кінотеатром та надано її основні переваги у комплексі роботи мережі цифрових кінотеатрів формату D-Cinema.

3 ОСОБЛИВОСТІ ВІРТУАЛІЗАЦІЇ МЕРЕЖІ ЦИФРОВОГО КІНОТЕАТРУ

3.1 Передумови переходу до нових технологій в кіно

Кінотеатри з часу свого створення як основний носій інформації використовували кіноплівку для архівування та демонстрації кінофільмів в кінозалі. З розвитком технологій склад та структура носія змінювалась, так наприклад з'явилась можливість записувати звук одночасно з зображенням. Але тут виникла інша проблема, яка була пов'язана з фізичним об'ємом розміщення носія. Так, за довжиною це була бобіна з кілометровою плівкою. Логічним виходом з цієї ситуації стали цифрові технології в кіно. Саме з них у дистриб'ютерів кіно робота стала набагато простіша, адже кінофільм який можна розмістити або на жорсткому диску, або навіть на портативній мобільній флешці не йде в ніяке порівняння з традиційними класичними підходами в кінотеатрі. З іншого боку, технологічний прорив у цифровому кіно дозволив створити кінопроектор, який має можливість відтворювати зображення без використання плівки. Враховуючи наявні недоліки кіноплівки – знос, об'єм, чутливість до температурних параметрів в приміщенні, де зберігається носій, цифрові технології в кіно дозволили обійти ці обмеження.

У 2 розділі дослідження була розглянута узагальнена структурна схема сучасного кінотеатру формату D-Cinema, і тут зовсім не буде відкриттям той факт, що усі складові цієї системи можна керувати на основі всесвітньої мережі Ethernet, адже у функціональному плані вони всі мають локальні фізичні порти та фізичні адреси, що їх ідентифікують при роботі в рамках однієї системи Digital Cinema. Разом з цим виникла ідея виробляти цифрові копії фільмів і доставляти їх в кінотеатри по всьому світу. Зрозуміло, що така реалізація потребувала б дотримання захисту інтелектуальних прав на ці кінофільми. Тепер фільми, окремі фільмотеки можуть зберігатись на невеликих серверах замість величезних складів з бобінами, а сама картинка

на екрані формується мікродзеркальними або рідкокристалічними матрицями з використання лазерних технологій, замість використання стрічкопротяжних механізмів з роліками, а нові фільми передаються на жорстких твердотільних дисках або за допомогою захищеного супутникового каналу, або на основі волоконно-оптичних ліній зв'язку через використання мережі Інтернет. І тут на передній план виступає питання захисту контенту, який передається в мережі кінотеатрів. Ще в 2005 році спілка 7 світових кіностудій випустила стандарти, які упорядковують вимоги до всіх частин сучасного кіно обладнання Digital Cinema з огляду на забезпечення останніми якісного зображення на кіноекрані та якісного багатоканального звуку в глядацькій залі кінотеатру. Така стандартизація дозволяє кіностудіям, виробникам кіно обладнання, дистриб'юторам та власне мережам кінотеатрів по усьому світі бути впевненими в повній сумісності всіх складових систем кінопоказу.

Однією з провідних завдань цієї спілки світових кіностудій було розроблення методу збереження фільмокопій. За результатами цієї розробки було видано чергову специфікацією, де цифрова мастер-копія, яка повинна була б надіслана в кінотеатр, в якому було організовано поєднання складових зображення (кіноряду), звуку (сигналогами) та текстової частини (субтитрів). З іншого боку, цей пакет повинен відтворюватись на будь-якому сучасному кіно обладнанні, і передусім на цифрових проекторах різних фірм-виробників, незалежно це Barco, Christie чи Kinoton. Іншою технічною проблемою відтворення цифрового пакету стала підтримка цифровим проектором роздільної здатності зображення кіноряду. Тут мова йде про те, що кіно копія може бути записана у форматі, наприклад, 4K, а сам проектор в кіноапаратній здатний відтворювати контент лише у форматі 2K. Зрозуміло, що не врахування цієї обставини може призвести при перегляді кінофільму до спотворень зображення і взагалі, може бути чинником до змазування суб'єктивних почуттів у глядачів кінотеатру.

Кінопакет являє собою систематизовану послідовну архітектурну ієрархію, яка складається з декількох папок, як і в будь-якій операційній системі. Цей підхід визначає прозорість складових пакету і дозволяє зберігати в собі різні види інформації. Це фактично “корзина” з складовими для файлів, яка може містити або не містити повний фільм. Тобто, цифровий кінофільм - структурований набір файлів з певним розширенням, які називають ще контейнером. Така структура може бути розширена шляхом додаткового розміщення, наприклад, трейлерів-анонсів кінофільмів і рекламних оголошень, як статичних так і динамічних. Все це, зрозуміло дуже зручно, адже на основі цих складових можна дуже швидко створити циклічний за часом роботи кінотеатру лист перегляду, але з іншої сторони, кількість таких допоміжних файлів може сягати десятки, особливо коли джерела надходження цього допоміжного контенту є різні. Архітектуру контейнера (не плутати з визначенням контейнеру при стисненні відеоданих) розробили та представили ще на початку 21 сторіччя і вона була дійсно революційною у розрізі існуючих форматів засобів масової інформації на той час. Характерною рисою функціонування кінокомпаній є ринок кіно послуг, який простягається по усьому світі і тут, можуть бути причини, які обмежують передавання кінокопії в кінотеатр. В даному випадку виникають чинники, які пов’язані не тільки з прокатною ліцензією фільму, але й слід враховувати ментальність людей, культурні традиції, мову спілкування, сумісність стандартів проекції. Тобто, приходимо до того, що кінокомпанія повинна враховувати ці чинники і фільм на сьогоднішній день може мати фактично індивідуальні версії. Наприклад, можуть бути версії з конкретним звуковим форматом, титрами, визначеним форматом зображення, на одному або декількох мовах. Це обумовлено обмеженнями кінопроектора та необхідністю показу фільмів на різних континентах. Архітектура контейнеру була розроблена для ефективного вирішення проблеми необхідності доставляти кілька версій фільму в кінотеатри, надаючи механізм для обміну файлами між ними.

Цифровий кіно пакет може мати у своєму складі і декілька контейнерів, або взагалі вирізнитися лише наявністю частини контейнера. Враховуючи це, пакет обов'язково повинен мати список опису складових, який називається Packaging List. Контейнер за архітектурою складається з декількох файлів, а саме списку відтворення і файлів доріжок. Для гнучкості кожен такий файл містить лише один тип сутності (Essence), такий як зображення, звук або субтитри. Спосіб та послідовність відтворення файлів вказано у списку відтворення, який називається CPL.

Файли доріжок визначаються окремою версією формату обміну даними MXF. MXF забезпечує фактично метод для перенесення різних типів складових кіно пакету з описом (з метаданими). Обмеження, що застосовуються до MXF-файлів для цифрового кінозображення і звуку, визначені в стандарті SMPTE ST429-3. Файли доріжок MXF складаються з заголовка, основної частини та нижнього колонтитула. Заголовок містить метадані, що описують файл доріжки. Нижній колонтитул містить таблицю індексів сутностей.

Цифрові потоки звуку та зображення оформлено з використанням технології Key-Length-Value. Саме поле сутності містить один кадр зображення або звуку. Один контейнер повинен мати за ієрархічним порядком мінімум три складові - композиційний список відтворення (Composition PlayList), файл доріжки зображення і файл звуку.

Крім того, ці доріжки можна розділити на кілька файлів, що складаються з частин цих доріжок. Така структура дозволяє з легкістю змінювати фільм, адже можна замінити лише окрему частину замість всього фільму, і це в кінопрокаті використовують, наприклад, для того щоб при кінопоказі вставляти в лист показу рекламні файли. Можуть виникати ситуації, коли контейнер може містити до 100 файлів або навіть і більше. Кількість файлів доріжок може бути обмежена алгоритмами шифрування контенту.

В контейнері може бути лише по одній сумарній доріжці зображення, звуку, звичайних і прихованих (для людей з вадами зору) субтитрів. Але при цьому, можна здійснювати обмін файлами доріжок між різними контейнерами. Наприклад, різні аудіовізуальні дані, що представляють собою різні версії фільму можуть бути призначені для показу в різних країнах, і вони можуть мати різні файли звукових доріжок, але при цьому визначатись лише одним однаковим файлом доріжки зображення. Аналогічно, можна мати дві контейнери з однаковим звуковим супроводом і різними доріжками 3D і 2D-зображення. Таким чином, можна передати багато версій фільму без необхідності відправки дублікатів файлів доріжок.

Таким чином, на даному етапі можна сформулювати наступні висновки з урахуванням модернізації кінотеатру до підтримки останнього “хмарної” платформи OpenStack:

- копії фільмів мають чітку ієрархічну структуру і оформлені на сьогодні на основі впроваджених стандартів у формі цифрових пакетів. Оскільки ці пакети дуже легко збираються та змінюється їх зміст, то є можливість перебудувати технологію доставки пакету в кінотеатр;
- не дивлячись на те, що рекомендації провідних кінокомпаній оформлені у формі відповідних стандартів, зусилля виробників кінопроекторів, серверів, звукових кінопроцесорів, серверів зберігання все одно спрямовані на унікальність своєї продукції. І це стосується не лише фізичного рівня інтерфейсів, але й унікального інтерфейсу, управління, контролю, і навіть способів проекції тощо. Цей фактор підштовхує до технічного рішення, яке стосується того, що сам процес проекції не повинен залежати від форми та наповнення цифрового кіно пакету, тобто повинна бути повна сумісність між технічною складовою відтворення та контентом;
- особливості, які пов'язані з ринком кіно послуг можуть бути причиною створення індивідуальних кіно копій з певними

обмеженнями. І оскільки є традиція та правило, що один кіно пакет – один кінофільм з своїми складовими – зображенням, звуком та субтитрами, то цифрові об'єми носія, особливо коли фільм у високому розширенні (4K та вище) з підтримкою 3D-кінопоказу, можуть бути значними. Тобто, мова йде про те, що кількість таких цифрових кінопакетів може визначатись кількістю країн, де відбувається кінопоказ і це число може вираховуватись десятками контейнерів. Іншими словами, логічно допустити, що було б зручно організувати певне місце, де всі ці версії знаходились. І доступ до цього місця за умов ліцензійних прав не обмежувався для власників мереж кінотеатрів.

- сучасне кінообладнання в переважній своїй більшості має мережеві інтерфейси, а отже управління, контроль і навіть пересилання контенту може здійснюватись на основі всесвітньої мережі Інтернет. Цей фактор може значно спростити роботу мереж цифрових кінотеатрів стандарту D-Cinema незалежно від того, де вони географічно розташовані. Реалізація такого підходу можлива не в останню чергу і через можливості платформи Open Stack.

У розрізі впровадження підтримки платформи Open Stack визначені вище особливості не враховують питання захисту контенту, а тому логічно розглянути далі в дослідженні особливості захисту складових кіно пакету, які сьогодні використовуються у сучасних кінотеатрах.

3.2 Особливості захисту кінофільмів в кінотеатрах

Окремим завданням спілки відомих кінокомпаній було створення методу та підходів захисту фільмів від не законного копіювання. Відомі на сьогодні два методи шифрування даних в кіно – симетричний та асиметричний. З них, за надійністю викликає довіру асиметричний підхід адже, при симетричному дані шифруються і розшифровуються одним і тим

же ключем. І цей ключ повинен мати кінотеатр, але тут виникає проблема передачі цього ключа. Тобто кінотеатр повинен мати захищений канал зв'язку, який можна організувати на основі, наприклад, “хмарного” простору, адже наприклад, такі сервіси, тобто онлайн-послуги, у формі розглянутих в першому розділі дослідження моделей SaaS, PaaS, IaaS як мінімум забезпечують за замовчуванням криптографічний захист даних – алгоритм розширеного шифрування AES-алгоритм.

Асиметричний підхід передбачає, що використовується пара ключів – публічний та приватний. Сутність підходу полягає у тому, що адресат (мережа кінотеатрів) генерує цю пару ключів. Відправник (наприклад, посередник кінокомпанії) отримає публічний ключ, на його основі шифрує інформацію, після чого розшифрування можливе тільки отримувачем і тільки його приватним ключем, який зберігається в секреті. Обидва ключі пов'язані між собою математичним законом, який працює лише в одну “сторону”. Тобто, на основі відкритого ключа не можна отримати приватний, адже фактично це 2 окремі частини єдиного ключа, який відповідає певному сертифікату-ліцензії на перегляд фільму. Приватний ключ в даному випадку може бути прив'язаний до конкретного серверу кінопоказу або до кінопроектору (обладнання), яке розташовано в кіноапаратній кінотеатру. Стандарт з захисту цифрових кінокопій передбачає захист фільмів за допомогою обох способів шифрування. Кожна фірма-виробник серверів кінопоказу для кожної конкретної моделі генерує 2 сертифікати за стандартом X.509, публічний і приватний, і маркує їх серійним номером. Приватний сертифікат вшивається у прошивку обладнання системи, а публічний можна отримати, надіславши запит до виробника. Спочатку кіно пакет з фільмом шифрується симетричним ключем, і вже цей ключ потім зашифровується асиметричним алгоритмом на основі публічного сертифіката серверу кінопоказу, де буде проводитись перегляд фільму.

Симетричний ключ кінопакету, який зашифрований асиметричним алгоритмом називається Key Delivery Message. Далі цей ключ надсилається в

кінотеатр електронною поштою і завантажується до серверу, де він уже на основі приватного сертифіката серверу розшифровує симетричний ключ від фільму. Цей ключ має часові рамки дії. При шифруванні контейнерів зашифровуються лише файли доріжок. Алгоритм шифрування, який використовується в цифровому кіно, – це симетричний алгоритм розширеного шифрування (Advanced Encryption Algorithm, AES). У цифровому кіно використовується 128-бітний ключ.

Оскільки, алгоритм розширеного шифрування співпадає з тим, що використовується в платформі Open Stack, то цей процес захисту даних можна автоматизувати. Тобто, можна зробити ситуацію, коли потрібні ключі від певного фільму будуть знаходитись у захищеному середовищі, доступ до якого буде чітко прописаний власником контенту і узгоджений з певною мережею цифрових кінотеатрів. Такий підхід, на мій погляд значно ускладнює роботу піратів ліцензійних фільмів.

Слід відмітити, що захист фільмів в кінотеатрі забезпечує медіа блок, який може бути у складі серверу кінопоказу. Він містить всі необхідні елементи обробки в межах захищеної зони для формування зображення і звуку, а також визначає елементи інтерфейсу за межами захищеної зони.

На основі наведених рекомендацій у п.3.1 та 3.2 магістерського дослідження далі наведемо алгоритм, який слід виконати аби провести модернізацію мережі кінотеатрів формату D-Cinema, щоб остання характеризувалась підтримкою “хмарної” платформи Open Stack.

3.3 Технічні рішення з впровадження платформи Open Stack в цифрових кінотеатрах

Оскільки Open Stack – відкрита платформа для проведення “хмарних” обчислень, то для нашого експерименту використаємо сервісну модель (певна “приватна хмара”), яка ототожнює інфраструктуру як сервіс (IaaS) адже для проведення модернізації мережі цифрових кінотеатрів необхідно

мати: контроль над мережею та контроль і права над об'ємом простору в системі зберігання даних. Натомість, сервісну модель PaaS, яка розташовується всередині IaaS використаємо для забезпечення контролю додатків, наприклад, тих, що стосуються розкладу плейлистів, які формуються та зберігаються в сервері кінопоказу. Також в цьому середовищі можна забезпечити підтримку додатку, який спрямовано на роботу системи управління кінотеатром (так званий TMS-клієнт). Ну і остаточно можна на базі цієї моделі сформувати перелік осіб кінотеатру, які мають право доступу до налаштувань обладнання кінотеатру. Це може слугувати додатковим заходом щодо захисту контенту від несанкціонованого копіювання. В результаті проведеного дослідження нам необхідно віртуалізувати мережу цифрових кінотеатрів залишивши при цьому основні функції роботи кінотеатру – проекція зображення, ретрансляція звуку, забезпечення роботи систем глядацької зали тощо.

Платформа Open Stack складається з підсистем, які можуть використовуватись автономно і їх опис ілюструє рисунок 1.3.

Отже, враховуючи, що сервіс платформи Open Stack (Nova) можна встановити на усіх обчислювальних вузлах мережі, і він керує складовими віртуальної машини (операційної системи), то можна його використати для генерування команд запуск, паузи та зупинки кіносеансу в мережі. А ці функції виконує в складі мережі одного глядацького залу (рис.2.1) – сервер кінопоказу на основі складеного плейліста (рис.3.1)

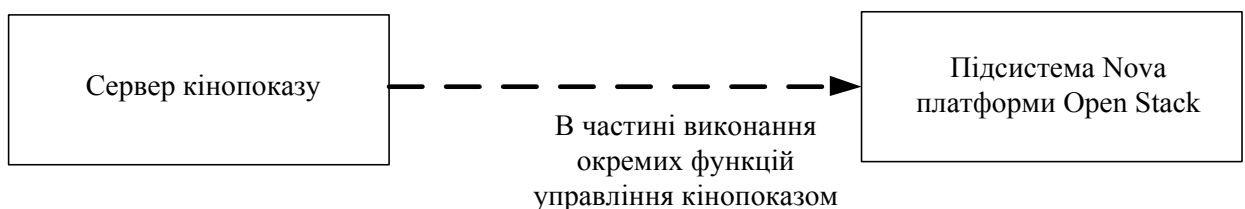


Рисунок 3.1 – Переналаштування роботи складових кінотеатру

Аналогічно, якщо використати верхній рівень мережі Digital Cinema (рис.2.2), то певні функції можна перенести для усіх серверів кінопоказу (рис.3.2). Плюсом такого підходу, є те, що можна організувати віддалене

управління кінопоказом і необхідність в знаходженні кіномеханіка в кіноапаратній в області виконання цих функцій просто відпадає. Крім цього, як вже було відмічено у другому розділі система автоматизації кіно забезпечує управління в глядацькій залі освітленням, завісою , а отже і ці функції можна віртуалізувати, тобто забезпечення передавання цих команд організувати віддалено. Зрозуміло, що такий підхід потребує синхронізованої роботи з сервером кінопоказом. А тому, рисунок 3.1 можна дещо розширити (рис.3.3)

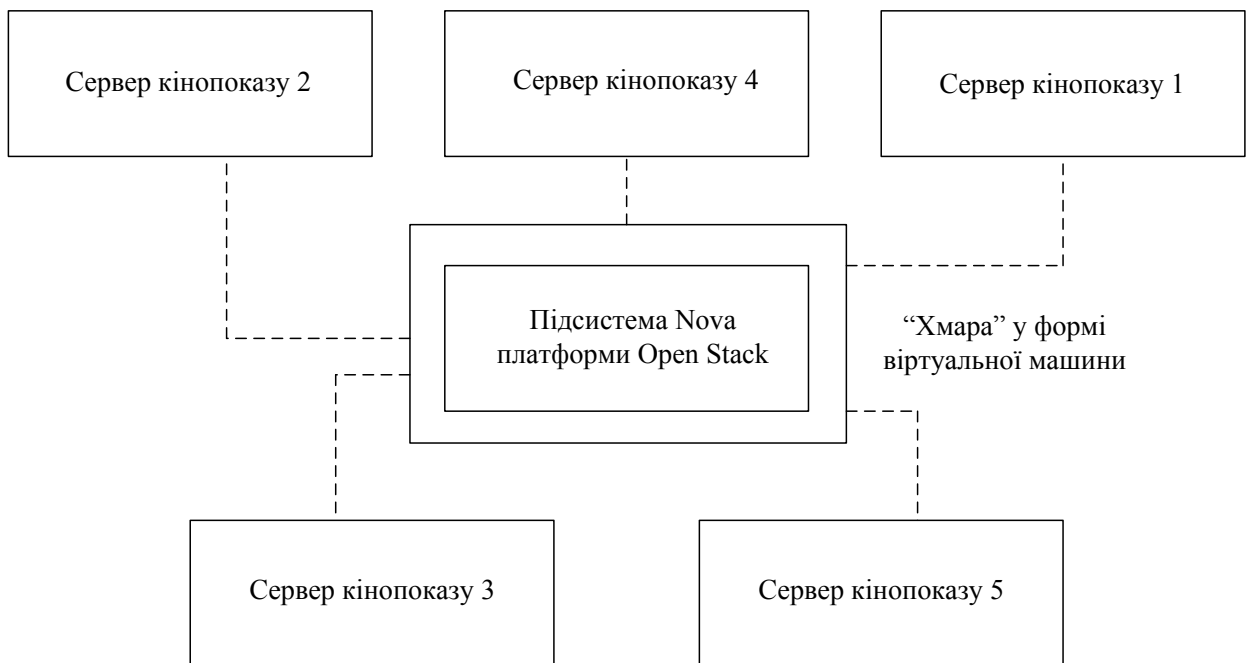


Рисунок 3.2 – Переналаштування роботи складових кінотеатру
(верхній рівень)

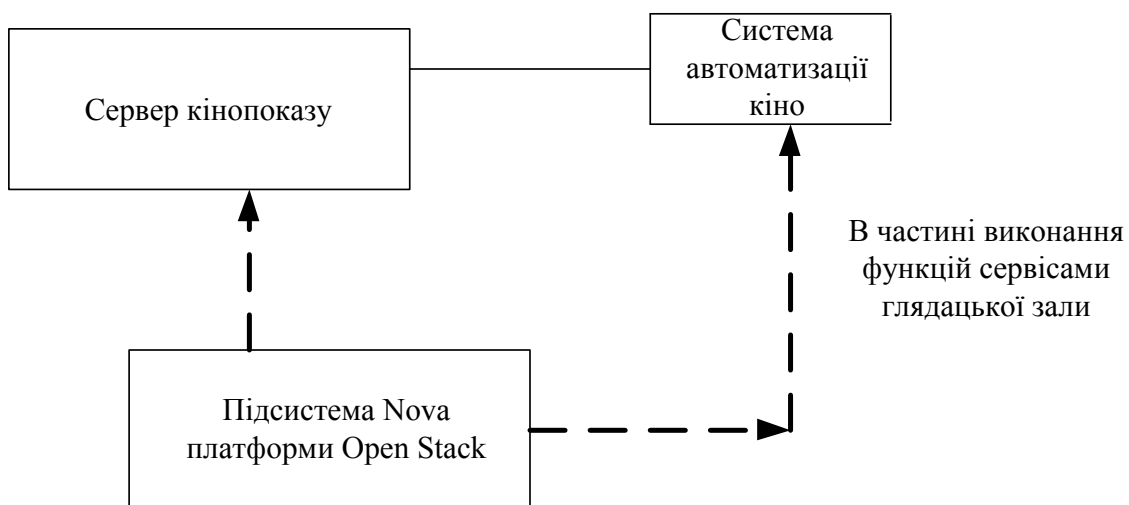


Рисунок 3.3 – Переналаштування роботи складових кінотеатру

Сервіс Open Stack Networking (Neutron) надає можливість для обладнання мережі Digital Cinema задати фіксовані IP-адреси. І це можна використати при проведенні діагностики обладнання або при першому запуску мережі Digital Cinema. Таку функцію можна запровадити для серверу кінопоказу, лазерного кінопроектора, кіно накопичувача, тобто в даному випадку для мережі кінотеатру з'являється можливість побудови чіткої схеми зв'язків між основним обладнанням глядацької зали (рис.3.4)

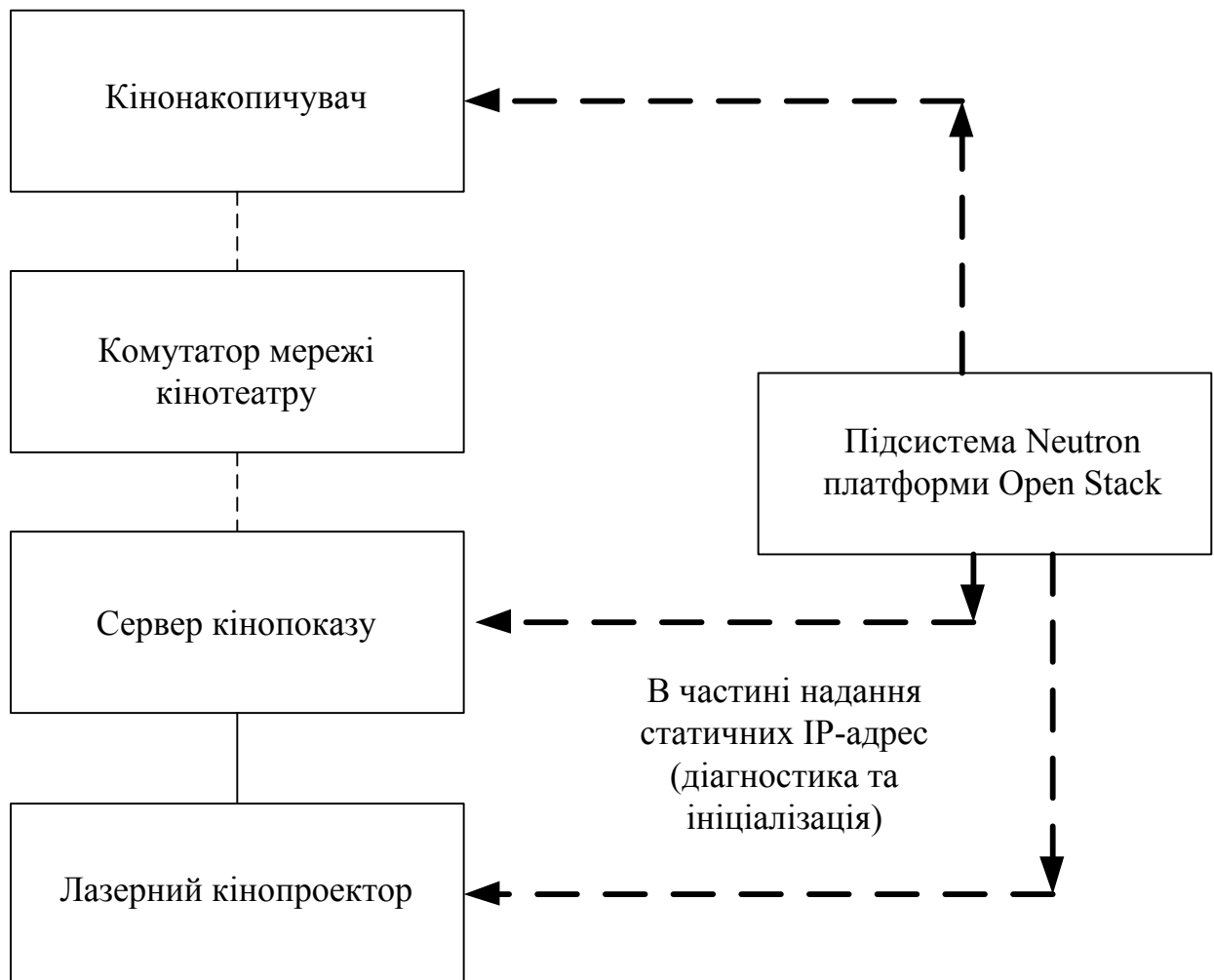


Рисунок 3.4 – Переналаштування роботи складових кінотеатру
(для окремої глядацької зали)

Служба ідентифікації Open Stack Keystone може бути використана в роботі мережі як система аутентифікації, тобто в ній може бути записано наявні на даний момент публічні ключі для захисту копії фільму. Зауважимо, що приватні ключі не можна перенести у віртуальний простір і тому дана

підсистема виконує в структурі кінотеатру лише допоміжну роль (рис.3.5).

Таким чином, розширимо рисунок 3.3.

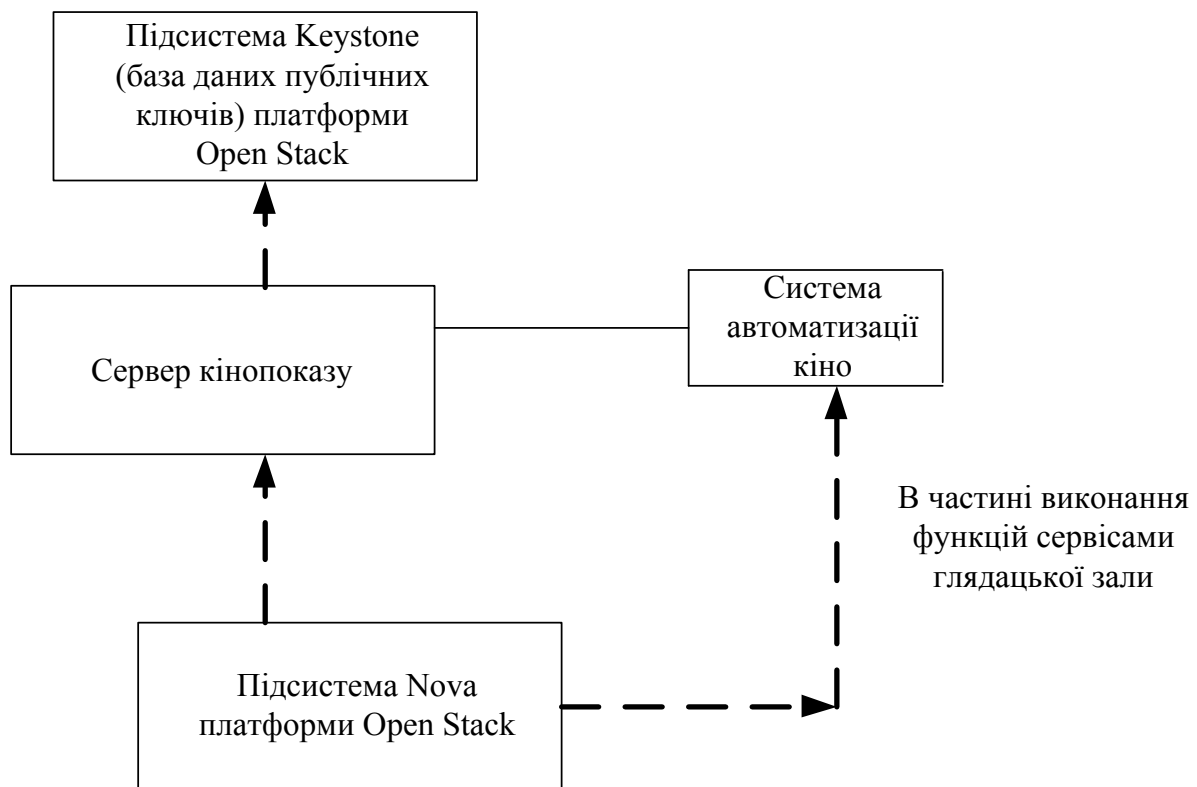


Рисунок 3.5 – Переналаштування роботи складових кінотеатру (захист даних)

Слід відмітити, що в представленій структурній схемі мережі Digital Cinema (рис.2.1) кіно накопичувач є окремим елементом і в даній технічній реалізації передбачається, що сервер кінопоказу не має у своїй структурі накопичувачів інформації, або має але вони не введені в дію.

Сервіс Open Stack Object Storage (Swift) можна використати як сховище даних і в системі Digital cinema ця підсистема може повністю замінити кіно накопичувач. Перевагою такого рішення можна вказати те, що підсистема Swift має розподілену архітектуру та надмірність а це у свою чергу значно підвищує надійність проведення кінопоказу в мережі кінотеатрів (рис.3.6).

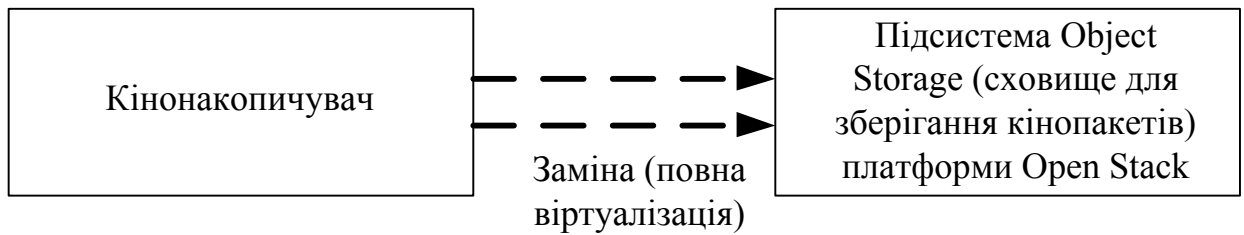


Рисунок 3.6 – Переналаштування роботи складових кінотеатру
(зберігання кінопакетів)

Сервіс Open Stack Dashboard (Horizon) відповідно до схеми на рис.1.3 забезпечує встановлення управління ресурсами мережі Digital Cinema. А отже в даному випадку цю підсистему у програмному режимі можна долучити для заміни TMS-клієнта, тобто пере налаштувати управління системою Digital Cinema (рис.3.7). Враховуючи що в даному випадку, мова йде про програму управління, яка в початковому режимі встановлюється у прошивці кіно накопичувача, то тут віртуалізація відбувається за рахунок фактично зміни місця розташування точки управління системою.

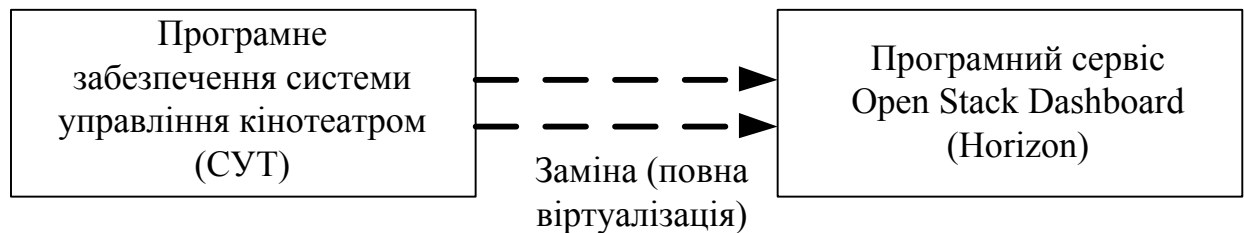


Рисунок 3.7 – Переналаштування роботи складових кінотеатру
(управління системою)

Таким чином, за результатами проведеного експерименту можна сформуванати змінену структурну схему мережі Digital Cinema (рис.3.8), яка була наведена у класичному варіанті на рисунку 2.1. Слід відмітити, що на схемі на рисунку 3.8, як і на вище наведених рисунках 3.1-3.5 штрих-пунктирна лінія відповідає Ethernet-з'єднанню, і усі елементи мережі мають вихід до зовнішнього сегменту мережі Інтернет.

4.1 Опис ідеї проекту

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Web سینما – це простий і зрозумілий сервіс для перегляду за допомогою мобільного гаджету кінофільмів, які знаходяться у хмарному середовищі кінотеатру	1. Перегляд кінофільмів.	Зручність перегляду вдома.
	2. Перегляд новинок.	Актуальність отримання інформації.
	3. Перегляд стрічки новин.	Персональний сервіс, доступний будь-якому клієнту 24/7.

Опис до таблиці 4.2:

W – слабка сторона;

N – нейтральна сторона;

S – сильна сторона.

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Web-TV	IPTV	SAT-TV	Opt- telecom			
1	Трафік за оплатою	+	-	+	-		+	
2	Технології	GSM, EDGE,	IP	DVB-S	IP			+
3	Час доступу	+	+	+	-			+
6	Об'єкти прийому	телефон	ПК	SAT- ресивер	ПК		+	
7	Контакт- центр	+	-	-	+			+
8	Вартість послуги (ключ)	10 \$	5 \$	20 \$	10 \$			+
9	Абон.плата сервісу	12.5\$/міс	3.85\$/міс	7.95\$/міс	4-15\$/ міс		+	

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Онлайн-кіно	Anroid	наявна	доступна
2	Доступ через мережу	SIM карта	наявна	доступна
3	Контакт- центр	«Контакт- центр по запросу» послуга 0-800	необхідно розробити	доступна
4	Персональний онлайн сервіс	Програмне забезпечення для ОС: Windows, Android, Mac	необхідно розробити	доступна
5	Автономне живлення	(Ni-Cd), Li-Pol, (Ni-MH), (Ni- MH) акумулятори	наявна	доступна

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 4.4 - Попередня характеристика потенційного ринку

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Динаміка ринку (якісна оцінка)	Зростає
3	Наявність обмежень для входу (вказати характер обмежень)	відсутні
4	Специфічні вимоги до стандартизації та сертифікації	DCI
5	Середня норма рентабельності в галузі (або по ринку), %	55%

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Розважальні заходи. Відпочинок дітей. Охоплення усіх категорій людей.	Родини, пенсіонери, молодь.	Залежно від цільової групи послуга комплектується різного роду додатками для зручності користування. Залежно від вподобань цільових сегментів, пристрій синхронізовано з різними ОС.	- надійність - зручність - доступність - простота - екологічність - швидкість

Таблиця 4.6 - Фактори загроз

№	Фактор	Зміст загрози	Можлива
---	--------	---------------	---------

			реакція компанії
1	Незацікавленість клієнтів	Внаслідок невдалого маркетингу клієнт може не зацікавитись послугами	Внесення додаткових сервісних послуг та зниження цін
2	Втрата монополії	Втрата рангу єдиного гаранту якості технології	Якісне та кількісне нарощування інтенсивності

Таблиця 4.7 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Монополія	Інноваційний тип послуг	Стандартизація на високому рівні
2.Локальний	Відсутність єдиного національного постачальника послуг	Окремий підхід до кожної локальної ділянки
3.Міжгалузева	Конкуренція з іншими галузями (постачальниками апаратної частини)	Необхідність співробітництва в окремих сегментах
4.Товарно-видова	Подолання розсинхронізації відбувається за схожими технологіями, що реалізовані апаратно	За необхідності, використання приладів схожого типу

Продовження таблиці 4.7 - Ступеневий аналіз конкуренції на ринку

5.Цінова	Можливість заощадити за	Гнучка політика цін на
----------	-------------------------	------------------------

	допомогою діагностики	доступ
6.Марочна	Кожна діагностика має бути стандартизованою	Отримання монополії над стандартом синхронізації

Таблиця 4.8 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Технологічні постачальники	Необхідність пошуку постачальників	Залучення малопопулярних постачальників	Незалежність у прийнятті клієнтських рішень	Надання переваги більш авторитетним технологічним рішенням
Висновки:	Незначна	Можливість виходу на ринок є	Постачальники диктують цінову політику на обладнання	Клієнти диктують вимоги до якості	Обмеження існують лише у разі відмови від діагностики

Таблиця 4.9 - Обґрунтування факторів конкурентноспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Раціональніший ціновий показник	Можливість більш раціонально використати ресурсів
2	Надання персональних сервісних послуг 24/7	Сервісна підтримка апаратної та програмної частини
3	Синхронізованість	Синхронізація з усіма ОС.
4	Спектр застосувань	Використання для ряду потреб користувачів.

Таблиця 4.10 - Порівняльний аналіз сильних та слабких сторін Web-cinema

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	1	2	3
1	Раціональніший ціновий коказник	13			+				
2	Надання персональних сервісних послуг 24/7	15			+				
3	Синхронізованість	20	+						
4	Спектр застосувань	17		+					

Таблиця 4.11 - SWOT- аналіз стартап-проекту

Сильні сторони:, надання персональних сервісних послуг 24/7, синхронізованість	Слабкі сторони: раціональніший ціновий показник
Можливості: використання для ряду потреб користувачів	Загрози: незацікавленість клієнтів, втрата монополії

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.12 - Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Молодь	Середня	Високий	Високий	Середня
2	Підлітки	Середня	Високий	Середній	Середня
3	Сім'ї	Середня	Висока	Низький	Середня
4	Пенсіонери	Висока	Висока	Середній	Середня
5	Web ресурси (сайти, клуби, соц.мережі)	Висока	Висока	Середній	Середня
Які цільові групи обрано: мережі магазинів та web ресурси					

Таблиця 4.13 - Визначення базової стратегії розвитку

№	Обрана альтернатива	Стратегія	Ключові	Базова
---	---------------------	-----------	---------	--------

	розвитку проекту	охоплення ринку	конкурентос- проможні позиції від- повідно до обраної альтернативи	стратегія розвитку
1	Створення гаранту якості державного рівня	Встановлення єдиного універсального стандарту	Розробка і випуск власних пристроїв	Стратегія диференціації
2	Дешевизна проекту	Раціональніші витрати на обладнання, та послуги	Маловідомі партнери з постачання обладнання	Стратегія лідерства по витратах

Таблиця 4.14 - Визначення базової стратегії конкурентної поведінки

№	Чи є проект «пер- шопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конку- рентної поведінки
1	так	Забирати існуючих та шукати нових	Характеристики програмного інтерфейсу	Стратегія виклику лідера

Таблиця 4.15 - Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока якість послуг	Стратегія диференціації	Синхронізованість	Якість, надійність, сервісність
2	Мінімальні витрати	Стратегія лідерства по витратах	Широкий спектр застосування	Дешевизна, раціональність, тех. підтримка

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.16 - Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Якість	Висока якість, сервісність	сервісність
2	Дешевизна	Раціональне використання коштів	дешевизна

Таблиця 4.17 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Дешевий якісний товар та послуги, стандартизована якість послуг та обладнання		
II. Товар у реальному виконанні	Властивості/характеристики:	М/Нм	Вр/Тх /Тл/Е/Ор
	1) Варстість обслуговування, 2) Кількість елементів	1) М 2) М	1)Е 2) Пр
	3) Строк безвідмовної праці 4) Технологічна собівартість товару	3) М 4) М	3)Нд 4)Тх
	Якість: дерстандарт якості, високоякісні технології		
III. Товар із підкріпленням	До продажу – діагностика, обладнання, кріплення, дод.елементи живлення Після продажу – персональний онлайн сервіс		

Таблиця 4.18 - Визначення меж встановлення ціни

№	Рівень цін на послуги замінники	Рівень цін на послуги аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	50-200 у.о./п	5-50 у.о./од	Середній	Н.10 у.о. – В.30 у.о.

Таблиця 4.19 - Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рек-ламного пові- домлення	Концепція рекламного звернення
1	Зацікавленість в якісному продукті з раціональним використанням ресурсів	Мережні ресурси	Синхронізованість з будь-якими ОС	Зацікавити у покращеннях пов'язаних із зростаючою популярністю товару та послуг	Представлення продукції відправною точкою на шляху до безпеки
2	Зацікавленість у великій кількості продукту із дотриманням умов якості	Мережні ресурси	Широкий спектр застосування	Зацікавити у позитивних сторонах	Представлення якісної роботи з клієнтами

Висновки

Комерціалізацію стартап-проекту Web-cinema після проведення детального аналізу, можна вважати доцільною. На дану пропозицію на ринку присутній значний попит з молоді, наразі він не задовольняється послугами заміниками. Рентабельність на ринку послуг насамперед обумовлена широким спектром застосування та персональним сервісом, доступним клієнтам 24/7.

Впровадження є перспективним, адже основними групами клієнтів є люди різних вікових категорій, клуби, сайти, соц.мережі. Конкурентноспроможність проекту обумовлена відсутністю аналогів з широким спектром застосування і наявністю лише товарів заміників, що, власне, і є основним бар'єром входження на ринок.

Обраною альтернативою впровадження було обрано - заключення договорів з мережами магазинів та інтернет ресурсами.

Імплементация проекту доцільна оскільки рентабельність та зацікавленість потенційних груп клієнтів створює досить сприятливі умови для розвитку проекту.

ВИСНОВКИ

В рамках магістерської дисертації було проведено дослідження з модернізації існуючої схеми мережі кінотеатрів Digital Cinema на прикладі обладнання однієї глядацької зали шляхом переведення мережі на підтримку відкритої платформи Open Stack. Зокрема, в рамках дослідження були отримані наступні висновки:

1. Визначено ключові особливості з впровадження нової відкритої платформи Open Stack у мережу цифрових кінотеатрів формату D-Cinema. Так знайдено, що для роботи елементів мережі цифрового кінотеатру доцільно використати дві сервісні моделі платформи - IaaS та PaaS, причому відповідні середовища-“хмари” повинні мати приватний характер.

2. Визначено, що за рахунок відкритості, гнучкості платформи Open Stack її можна адаптувати до будь-якої системи, яка має певні послуги, сервіси та додатки і характеризується виходом свого обладнання в мережу Інтернет. Разом з тим, відмічено, що захист даних, що передаються між підсистемами платформи визначається алгоритмом AES, що заслуговує довіри з сторони розробників.

3. Проаналізовано основні особливості зі створення схеми мережі Digital Cinema, розглянуто функції основних складових цієї системи. Зокрема, знайдено, що один сервер кінопоказу рекомендується використовувати для обслуговування не більше 3 глядацьких залів. Відмічено, що оскільки окремі функції роботи мережі які визначають управління глядацькою залю є типові, то їх можна автоматизувати або забезпечити передачу команд управління роботою віддалено.

4. В практичній частині дослідження було надано детальні рекомендації, які дозволяють провести модернізацію типової схеми мережі цифрових кінотеатрів шляхом впровадження окремих підсистем відкритої платформи Open Stack. Відзначено, що наприклад підсистема Keystone може відіграти при роботі серверу кінопоказу лише допоміжну роль, адже асиметричний алгоритм шифрування контенту, який на сьогодні є найбільш розповсюдженим обмежує повне використання цієї підсистеми.

5. На основі результатів проведеного експерименту знайдено, що деякі елементи системи Digital Cinema можна віртуалізувати повністю. Так в даному випадку, мова йде про кіно накопичувач та клієнтський профіль TMS. Таке технічне рішення дозволяє не тільки підвищити надійність роботи кінотеатру, а й дозволяє ввести додатковий захист кінофільмів, які відтворюються в кінотеатрі. Загалом отримана віртуалізація системи Digital Cinema через впровадження компонент платформи Open Stack склала приблизно 60% і цей показник теоретично можна ще підвищити через впровадження додаткових модулів відкритої платформи Open Stack.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Маркелов А. OpenStack: практическое знакомство с облачной операционной системой. Москва, 2006. 160 с., іл. ISBN: 978-5-97060-328-4.
2. Kinoton Special: "Theatre Management Systems: Networked Movie. URL: <https://www.yumpu.com/en/document/view/8274655/kinoton-special-theatre-management-systems-networked-movie> (дата звернення: 19.09.2018).
3. Единый сервер для кинотеатра с любым количеством зрителей. URL: <http://fbeed.ucoz.ru/news/cinecloud/2014-03-12-202> (дата звернення: 10.10.2018).
4. Савкова Н.В Мультиплекс. Хабаровск, 2013. 59 с., іл.
5. Самохин В. П. От камеры обскуры до цифрового кинематографа / за ред. Е. А. Тихомирова, В. П. Самохин / Наука и образование. – 2014. – № 11. – С. 13–53.
6. Шпула О. А. Реалізація хмарних технологій у сучасному проектуванні цифрових кінотеатрів *Elconf-2018*. – 2018. – Секція № 7. – С. 390–393.
- 7 Шпула О. А. Хмарні технології для проектування сучасних кінотеатрів на базі Openstack. *Сучасні проблеми застосування електронних та інформаційних технологій в телекомунікаціях, телебаченні та цифровому кінематографі: тези доповідей*. – 2018. – Секція А. – С. 39.

ДОДАТОК А

ABSTRACT

In the context of the rapid development of entertainment services in the world, almost the first place is the question of the creation of a cinema complex, where besides the digital auditorium and the traditional movie show, visitors can

also enjoy other entertainment activities - bowling, karting, ice rink and the like. In the meantime, the issue of multi-channel operation also contains technical difficulties, which are related to the provision of movie licenses for movie rental. Thus, it can be argued that movie distributors are no longer fully entitled to trust the owners of cinemas in terms of providing the latest film protection, especially when the premiere of this picture takes place around the world. That is, the issue of the protection of intellectual property rights does not play today a second role in the demonstration of motion pictures. The introduction of the OpenStack concept, which allows you to separate user services (viewer, cinema engineer, engineer) and product owner, can be a way to enhance content security.

Cloud computing is a model for providing widely available, easy access to the network to a common pool of computing resource requirements on demand. These resources include: networks, servers, storage systems, applications and services. Since the modern digital cinema, according to the scheme of connections of the main components is actually the information network, it is quite clear that such a model of calculations can be introduced into the robot of this network. There are three main cloud service models within the OpenStack concept: SaaS (Software as a Service - service as a service), IaaS (Infrastructure as a Service - infrastructure as a service) and PaaS (Platform as a Service - platform as a service).

Infrastructure as a service (IaaS) is a service model of OpenStack, which is used mainly for the deployment of clouds. In this case, the user receives control of all levels of the software stack lying above the cloud platform, namely: virtual machines, networks, allocated by the user in the amount of space on the data storage system (FDI). In this case, the user acts as the administrator of the operating system and everything that works on top of the applications. Examples of platforms that provide a similar model, in addition to OpenStack, include Apache CloudStack, Eucalyptus, and OpenNebula.

Platform as a service (PaaS) - a cloud built on such a model, may well be located "inside" the cloud of the IaaS model. In this case, the user limit is at the level of the platform for building applications, such as application servers,

libraries, development environments, or databases. The user does not control or administer the virtual machines and operating systems installed on them. Examples of PaaS cloud platforms: Apache Stratos, Cloud Foundry, Deis, and OpenShift Origin.

Software as a service (SaaS) - in this case, the control limit of the user - the application itself. In this case, the user may not even know what a virtual machine or operating system is, it just works with the application. Examples of such cloud products: Google Docs, Office 365.

For each model there can be four types of implementation of the "cloud": public (public); private (private); community for the specific community of consumers; and hybrid (hybrid cloud).

Private cloud - the entire infrastructure is deployed in the data center (data center) and serves as a subdivision of one company or group of companies.

The public cloud - the customer of cloud services can be any company or even a private person. This is an implementation model that earns cloud service providers.

In a public cloud, IT services and infrastructure resources are provided over the global Internet. The ability to use the third-party infrastructure creates many opportunities for efficient resource downloads and redistribution. The big disadvantage is that public clouds are much more vulnerable than private ones.

Because application servers are in a public cloud, the risk of simple business processes through the server crashes is eliminated. Virtual ISPs are most often configured on a powerful physical database, located in large datacenters, where downtime can be calculated in minutes per year. The use of public clouds and the lack of user contact with complex computer equipment allows you to abandon the services of IT professionals in a specific organization.

Instead, the main disadvantage of the public cloud is the lack of control capabilities on the part of the organization: serviceability is completely subordinated to the third-party provider. In addition, the performance of public cloud services depends directly on the stability and bandwidth of the Internet

connection, in some cases, data transfer can be quite slow. When operating large volumes of data public clouds can not withstand competition with private in productivity. Weak security is another feature of public cloud environments. Whatever efforts the security provider has made, the protection of the private cloud will always be more reliable. And this circumstance limits their use in the deployment of modern digital cinema format D-Cinema.

In a private cloud, services and infrastructural resources are interconnected based on a private network. This model guarantees a significantly higher level of security and control, but the cost of software and hardware when using this approach is significant compared to public clouds. For a private cloud, comparatively more control options than a public cloud, simply because all components of an IT infrastructure remain on the organization's side. Companies can monitor and manage cloud environments more efficiently. • A high level of security is ensured by the fact that the customer has one single organization of the service, so that the whole infrastructure can be optimally configured to meet existing security requirements.

The OpenStack project, also called the "cloud" operating system, consists of a number of separate projects for which separate open subsystems have been developed. A particular modification of OpenStack may include only a part of them. Some subsystems can be used altogether autonomously, or as part of other OpenSource projects. The set of services increases from version to version of the OpenStack project both due to the emergence of new ones, and through the division of the existing functional. Each project has its own documented set of REST APIs, command line utilities, and native Python interfaces that provide a set of functions similar to the command line utilities.

As with any operating system, OpenStack has users. After authentication, the user gets a token to access some or other resources. The user is created not only for the person who works in the system, but also for the services. Each service has one or more endpoints. The login point is the URL that this service is available to.

Token is a string of letters consisting of letters and numbers and is intended to access APIs and resources. Token is issued for a limited time and can be withdrawn if necessary until expiry. In order for the user to receive a token, he must either provide the name and password or the name and key for access to the API (API key). The token also contains a list of roles that determine the credentials available to the user.

Users and other resources are merged into projects (the documentation also uses the term tenant in English). A project is a container that can combine the resources of an individual organization that uses the public cloud of OpenStack, a separate application, or an individual user - you decide it yourself. In turn, projects are merged into domains. A domain is the largest container and domains define the namespace - the scope of the objects. For example, users should be unique within the same domain. By default, Keystone creates a domain named Default.

One of the basic services is OpenStack Compute (Nova). This service is installed on all computing nodes of the cluster. It provides a level of abstraction of virtual equipment (processors, memory, block devices, network adapters). Nova provides the management of instances of virtual machines, referring to the hypervisor and giving commands such as their start and stop. It's important to note that OpenStack technologies are hypervisor-independent. Support is implemented through the appropriate drivers in the Nova project. Development and testing of OpenStack are conducted primarily for KVM (Kernel-based Virtual Machine - a virtualization software). Most implementations are also linked to the KVM hypervisor.

Apart from managing virtual machines, part of Nova's services can also provide network management. Among the currently deployed clouds, the percentage of use of the nova-network service is quite high.

The following service called OpenStack Networking (Neutron) is responsible for network connectivity. Users can independently create virtual networks, routers, assign IP-addresses. These functions can be used when reconfiguring the digital cinema network, since elements such as a digital cinema

projector, a movie show server, an IMB block, and a TMS system are organized in the form of an information network.

It should be noted that one of the mechanisms provided by Neutron is called "floating addresses". Thanks to this mechanism, virtual machines can receive external fixed IP addresses. Through the mechanism of modules, you can implement a functional such as a balancer of network load as a service, a firewall as a service and VPN as a service. OpenStack Identification Service Keystone is a centralized directory of users and services to which they have access. Keystone acts as a single system of authentication of the "cloud" operating system. The service supports several types of authentication, including token authentication, using a username / password pair, and AWS-compliant authentication. Keystone supports integration with existing directory services, such as LDAP. In addition, Keystone is a directory of services available in OpenStack, supporting a directory of requisites for accessing the APIs of the corresponding services, and implementing a role-based access control policy. Keystone checks the validity of user accounts, checks the mapping of users for OpenStack projects and roles, and if successful, issues a token to access other services. Keystone also maintains a directory of services. This item can be used when planning cinema display playlists and when creating an access database to a digital cinema system.

OpenStack Image Service (Glance) maintains a directory of virtual machine images that users can use as templates to run instances of virtual machines in the cloud. Also, this service provides a backup function and the creation of snapshots. Glance supports many different formats, including vhd, vmdk, vdi, iso, qcow2, ami, etc.

The OpenStack Block Storage (Cinder) service manages a block repository that can use running instances of virtual machines. This is a permanent information store for virtual machines. You can use snapshots for storing and recovering information or for cloning. Often, cinder uses a Linux-based repository, but there are modules for hardware repositories.

The OpenStack Telemetry (Celiometer) service is a central source of information on clouds and monitoring data. This component provides billing capabilities for OpenStack.

OpenStack Orchestration (Heat) - a service whose task is to provide a lifecycle of applications in a cloud infrastructure. With the AWS CloudFormation template, the service manages other OpenStack services, allowing you to create most types of resources (virtual machines, volumes, floating IPs, users, security groups, etc.). Heat with Ceilometer data can also auto-scan applications. Templates describe the relationship between resources and this allows the Heat service to call the OpenStack API in the correct order, for example, first create a server, and then connect multiple volumes to it.

Lastly, the OpenStack Dashboard (Horizon) is the closest to the cloud user, which allows you to manage cloud resources through the web console.

Swift has a distributed architecture, providing horizontal scaling, as well as redundancy and replication for fault-tolerance. Swift focuses primarily on static data such as virtual machine images, backups and archives. Swift is a software-defined storage (SDS) that works with objects. Object repository, unlike file or block, provides access not to files and block devices, but to objects in a single namespace. The object repository has its own API, and usually objects are accessed via HTTP. Such a repository abstracts objects from their material position and allows you to scale independently of the physical infrastructure lying under the repository. Also, the advantage of an object repository is the ability to distribute requests for a large number of servers that store data.

The main features of Swift Architecture are: linear scalability; absence of exclusive roles or masters of operations; mechanisms of replication and self-healing. By default, each object has three replicas. Accordingly, you do not need a RAID array on the network storage nodes; support for large objects (by default up to five gigabytes). The Swift architecture provides "end-to-end consistency" (the English-speaking term - eventually consistent). This term means that in the absence of data changes, eventually all requests will return the latest updated

value. That is, when you load an object into a cluster, you will receive a confirmation that the object is recorded when the confirmation from the two nodes comes. The rest of the nodes will get the "end-to-end" object.